Application of Pattern Recognition Techniques for Early Warning Radar (EWR) Discrimination

Final Report
29 March 1995

Small Business Innovative Research Project

Supported by
the Ballistic Missile Defense Organization

Mr. Thomas Foster
Principal Investigator

SenCom
CORPORATION

Contract No. N00014-94-C-0240
SBIR BMDO 94-003
CDRL No. 0001AC

SenCom Corporation
3 Preston Court
Bedford, MA 01730

Research supported by Ballistic Missile Defense/Innovative Science and Technology and managed by the Office of Naval Research

Executive Summary

The purpose of this research was to develop and evaluate an ensemble of simple data processing and pattern recognition techniques that could provide the Early Warning Radars (EWRs) with an enhanced target discrimination capability.

The PAVE PAWS and the Ballistic Missile Early Warning System (BMEWS) radars are phased array radars which support the Early Warning System (EWS). These radars operate at UHF and were designed to detect and track large numbers of objects, as part of the then perceived threat, i.e., a massive ballistic missile attack of hundreds of ICBMs/SLBMs. Discrimination and tracking of individual objects was not optimized in their design. Our aim was to examine discrimination methodologies that could be implemented without modifying the EWR hardware. Any enhancements would be installed by software upgrades only.

Pattern recognition techniques represent one approach to extracting additional discrimination information from the radar cross section (RCS) data that is currently generated by the radar. For example, given approximately 100 seconds or more of radar cross section (RCS) data displayed as a function of time, the global signature of the first or second stage of the missile, or "tank object", results in a visual pattern that is very different from that of a reentry vehicle (RV). Moreover, there may be other structural components to the various patterns that could be used as discriminates. Our basic objectives were to develop and assess the mathematical techniques to extract these structures in a systematic way and to implement these techniques in a computer program.

To this end we surveyed, in a systematic way, the RCS times histories for a large number of objects. This data was from actual objects seen by a PAVE PAWS radar and not a simulation. A number of data processing techniques were developed to identify patterns within these time histories. The survey indicated that the techniques developed were useful in deriving various patterns from the data, and that differences in the patterns corresponded to different objects. This property could then be exploited to discriminate between objects. These techniques were then incorporated into a functioning classifier.

The work accomplished during phase I of this research project has produced a number of solid results.

First, we found that different types of patterns exist in the EWR RCS data base and simple processing techniques can be developed to identify the various aspects of these patterns. As discussed in the report, the data was collected from events that occurred over a period of about two years and thus the patterns are not the result of special circumstances. Thus there is real value in attempting to discriminate on the basis of these patterns.

Second, it is also shown that most of the RCS data sets fall naturally into a reasonable number of distinct pattern classes. That is, the number of pattern classes is much less than the number of data sets examined and that the members within a pattern class do look alike.

Third, we have demonstrated that different object classes tend to lie in distinct pattern classes. By this we mean that objects that we believe are RVs and tanks have different patterns. Thus being able to associate a pattern or pattern class to an object, i.e., doing discrimination on the basis pattern recognition, appears very possible.

Finally, a computer program has been developed which, by using the various methodologies developed, can generate pattern classes and perform a discrimination function based on the assignment of RCS time histories to these pattern classes.

These results strongly suggest that further work in this area will be very valuable. We intend to continue this work and move toward an evaluation and test phase.

Table of Contents

Section 4
Prototype Classifier Description

Section 5
Classifier Processing Results

Section 6
Summary Discussion and Conclusions

List of Figures

List of Tables

Section 1
Introduction

1.1 Purpose

The PAVE PAWS and the Ballistic Missile Early Warning System (BMEWS) radars are phased array radars which support the Early Warning System (EWS). Collectively these radars are referred to as the Early Warning Radars (EWRs). These radars operate at UHF and were designed to detect and track large numbers of objects, as part of the then perceived threat, i.e., a massive ballistic missile attack of hundreds of ICBMs/SLBMs. Discrimination and tracking of individual objects was not optimized in their design.

In our recent work over the past three years in support of the Ballistic Missile Defense Organization (BMDO), we have noted that the radar data generated by a PAVE PAWS radar might contain information that could be exploited by using pattern recognition techniques. It is our purpose to study and present techniques by which this information can be extracted and exploited against today's perceived threat, which is a limited ballistic missile attack, during which the discrimination and tracking of individual objects is a prime requirement.

Our fundamental objective during phase 1 of this research project was to assess the feasibility and application of simple pattern recognition algorithms and techniques to the problem of radar target discrimination and classification at UHF frequencies. A future goal would be to incorporate these algorithms into the EWRs as a low cost and low risk software upgrade to improve discrimination. To do this, we divided the problem into two aspects. We first determined some suitable ways in which to present the data prior to applying pattern recognition techniques. To a large extent this involved using MATLAB to survey a large portion of the PAVE PAWS radar cross section (RCS) data base. Based on these results, we then developed a prototype classifier based on simple pattern recognition techniques.

1.2 Background

The idea which we have exploited is that when the RCS amplitude is expressed as a function of time, we see a global structure that can be viewed as a pattern. Simple physical considerations suggest that different classes of objects will produce different patterns in the RCS. For example, tanks (first or second stages) are large objects and generally have a tumbling motion associated with their trajectory. This produces patterns that exhibit large means and variances. Just the opposite is true for reentry vehicles (RVs). In this case we are considering small objects which tend to be spin stabilized. This implies patterns with small means and variances. While such differences might not be apparent within a small sampling of data points, such patterns should and do manifest themselves over a period of hundreds of seconds.

Of course the reality of the situation is more complicated. We also need to consider the situation where RVs may be executing an effective tumbling motion, relative to the radar line of sight. Moreover, we realize that there are other objects that could approximate the size of the RV, and a more detailed look at the RCS pattern may be necessary to distinguish these cases.

1.3 Additional Technical Issues

As stated above, the EWRs, particularly the PAVE PAWS, were not designed for target discrimination. In fact these radars have a number of operating characteristics that make discrimination rather difficult.

The primary problem is that the EWRs transmit pulses that have a wavelength comparable to the length of an RV and other similarly sized objects. This means the small scale structure which would distinguish the RV from the nose fairing, for example, cannot be effectively sampled by the PAVE PAWS radars operating at UHF. (This ignores the possibility of operating the radar in a nonstandard mode and applying additional processing to the output as XonTech does during their radar tests.)

The question becomes how to distinguish between various classes of electrically small objects. There are two ways in which that might be done. First, one can postulate that over a long period of time, perhaps several hundred seconds, the changing aspect or viewing angle will induce visible changes in the RCS pattern that will distinguish the RV from other small objects. Secondly, it may be that while the operating wave length is relatively insensitive to the precise shape of the illuminated object, the various RCS patterns will be different in small details which can be examined by considering the higher moments of the distribution of RCS returns.

A second problem is the narrow bandwidth of the radar. This results in a relatively poor range resolution capability. For example, the PAVE PAWS radars have a one MHz bandwidth, which implies a 300 meter range resolution cell. Without additional information such as Doppler or possibly more sophisticated track processing, "misassociations" of closely spaced targets are likely to occur. This will result in data from two or more objects being assigned to the same track file.

A final problem concerns the track update rate. For the EWRs, the effective update rate is generally very low; often between one and one quarter Hz and sometimes even lower. In addition, one often finds that the track rate, whatever it is, may not even be constant. This makes the use of any kind of spectral analysis difficult at best. Thus any information concerning the rotation or tumbling motion of a particular object is likely to be severely degraded or difficult to interpret, at least over the short term.

From the point of view of pattern recognition techniques, these radar characteristics translate into three problems. First does the radar data have the fidelity to distinguish all the different patterns? That is, do we have a different pattern in the RCS data for each class of object that comes into the view of the radar? Or will nose fairings and RVs and other similarly sized objects all be mapped into the same basic pattern class? Secondly, with the inconsistent track update rate, we face the possibility that many data sets might not have a sufficient number of points to form a pattern, even if the set extends over a long period of time. Lastly, given the relatively poor range resolution of the radar, will misassociations create spurious patterns that can not be related to any specific object?

These are significant issues. However there is one operational aspect that works in our favor, and that is time. In a typical National Missile Defense (NMD) scenario we are not forced to make a classification decision within tens of seconds as we might if we were considering a Theater Missile Defense (TMD) situation. In fact we could have as much as eight hundred seconds in which to make our final classification decision. Even in the case of a sea launched threat, we would still have about 100 seconds before a classification decision was required. Thus we believe that sufficient time exists to allow the effects of misassociation to be sorted out. In addition, given that enough data points can be collected, we believe that distinct object classes will map into distinct patterns.

Moreover, given enough time, the effect of the non-uniform tracking rate should not pose an overriding difficulty. This is because as long as the set of non-uniform time samples does not match any integer multiple of the tumbling period of the object, we should always be seeing a different aspect of the body.

1.4 Outline of Report

As this report covers a great deal of material, it is useful to discuss the main activities and results that are presented in the following sections. The main idea is to start with actual PAVE PAWS data and attempt to identify patterns within the RCS Vs time histories. To this end, we begin by considering various ways in which to process and simplify the data. This is discussed in some detail in section 2. Also included in section 2 is a description of the data file structure and formats.

In section 3 we discuss the results of applying the data processing techniques to the data. Given the large number of data sets to examine, we discuss how these techniques were incorporated into a MATLAB script to allow for a systematic survey of the data. Some illustrative results from this work are presented.

In section 4 we discuss the function and architecture of the prototype pattern recognition classifier. This is an actual piece of working software and should be considered as a major product of this research.

The prototype classifier is designed to operate in one of two modes and this is discussed in section 5. In the clustering or training mode, the classifier accepts a large number of data sets and through various comparison techniques divides the data sets into a natural set of pattern classes. In the classifying mode, the program accepts a single data set and attempts to assign it to one of the established pattern classes.

Preliminary results from the clustering and classifying modes of the prototype classifier are presented in section 6. Section 7 summarizes the major conclusions and indicates the direction for future work.

Section 2
Data Processing Techniques

2.1 Description of Data

For our research, we have used data sets that represent the RCS time histories of selected targets seen by a PAVE PAWS radar at various times over a period of about two years. These radars normally output a considerable amount of tracking and system performance data in real time, most of which are written directly to files and stored. The information contained within these files can be accessed by specifying a particular Logical Record Identifier or LRID. There are a number of LRIDs that contain different combinations of information pertaining to the tracking of various targets that come into the radar's field of view. For our purposes, the data contained within the LRID 94 series are of most interest. This information can be down loaded by specifying the times of interest (i.e., the times corresponding to an interesting launch) and this was done to obtain our data.

The files are written in an ASCII format and contain a number of data fields that must be read. For example the data sets contain, along with other information, the track identification (ID) number; the time; signal to noise ratio (SNR); RCS (in dBsm); and the X, Y, and Z position of the object in radar face coordinates. The file is structured in a time sequence relating to when the information pertaining to a particular object was written into the file. Thus the various track ID numbers and their RCS values will be interwoven within the file. However it is a trivial matter to write a read routine to select out any particular object from the track file and collect its RCS time history. In fact this was done initially to obtain sample data sets to analyze during the earlier phases of this study.

Given the number of objects contained within the track files, it soon became necessary to develop methods to systematically search through the track files and select out those data sets that satisfied certain criteria such as a minimum track length, adequate number of data points and a reasonable track update rate. This was also done and resulted in the generation of data sets that were discussed in Progress Report II (PR II).

However, during the actual operation of the radar, the "off line" selection of specific data sets would not be practical. Thus whatever classifier is developed must be able to read a portion of the LRID track file directly. To simulate this, we combined all of the track files into one large LRID-like file. For our situation, certain fields were dropped such as the SNR, since these values could give insight into classified areas of radar sensitivity. For all of the testing and development of the classifier, this LRID file was used.

A sample of this file and the data fields is shown below in table 1.

6147	405.10	-10.000	511319.60	467677.60	1439607.00
6148	405.20	2.305	-2518726.00	466506.90	1747672.00
6147	405.31	-10.000	510658.60	467539.80	1440325.00
6146	405.31	1.777	-1570388.00	1197227.00	3913717.00
6150	405.64	4.878	506736.40	463463.70	1439534.00
6149	405.64	4.260	485955.30	458689.40	1458822.00
6151	405.64	8.059	-2468564.00	476799.30	1808043.00
6151	405.86	7.041	-2468902.00	472527.40	1811644.00
6145	405.96	-12.694	507949.40	462766.60	1439918.00
6150	406.07	-3.353	503964.10	461973.20	1441792.00
6149	406.07	-3.519	482691.30	455001.30	1461872.00
6151	406.18	8.054	-2469202.00	474396.20	1813706.00
6150	406.29	13.695	503361.40	463378.50	1441990.00
6149	406.29	-2.001	481459.20	454314.10	1463123.00
6148	406.61	-.241	-2518726.00	466506.90	1747672.00
6147	406.61	-10.000	507622.70	466913.90	1443620.00

Table 1

The first field or column gives the track identification number (ID) and one can see how the track IDs are interwoven throughout the file. The second column is the time in seconds and these are all referenced to a particular event. The third column gives the RCS in dBsm and the last three columns give the X, Y, and Z position of the object in radar face coordinates. One should note that in the actual track LRID, the time would be given in hours, minutes, seconds and milliseconds. The transformation to just seconds is done for our convenience.

2.2 Description of Processing Techniques

As stated previously, when the RCS data is plotted as a function of time a finite number of basic patterns emerge. As examples one can consider the RCS versus (Vs) time plots shown in figures 1 and 2.

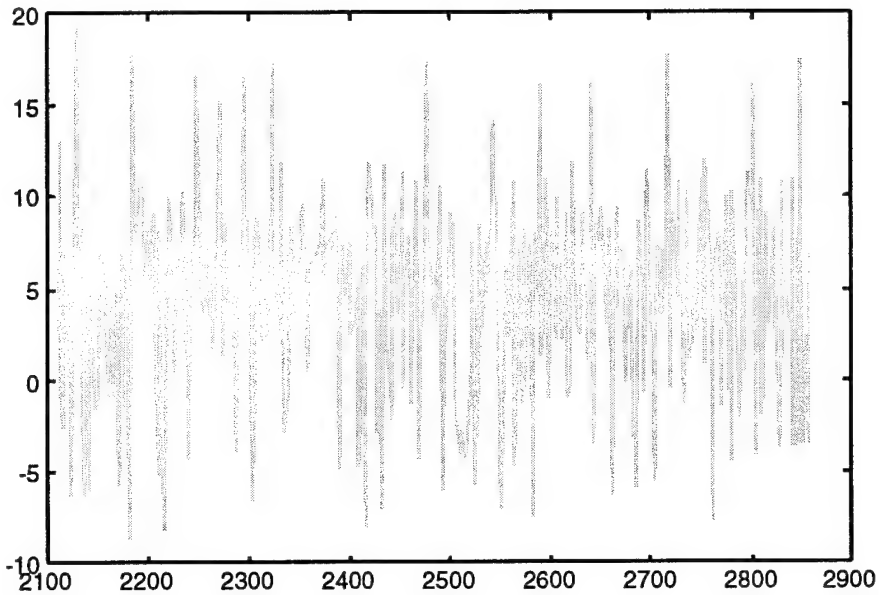

Figure 1. RCS Vs Time (Object 2348, Tank)

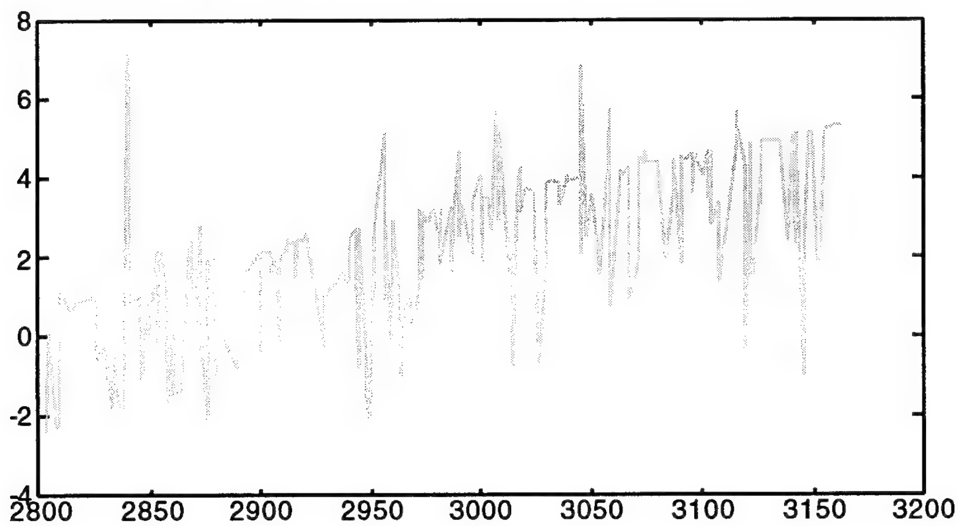

Figure 2. RCS Vs Time (Object 2609, RV)

It is hypothesized that the different patterns, such as those shown above, represent different types of objects such as tanks, RVs, post boost vehicles (PBVs) or fragments. However, these patterns also have a great deal of structure, some of which may be difficult to interpret or even misleading. The first step then is to develop a number of processing techniques that simplify the data while enhancing certain features indicated in the original patterns. We will denote the result of each of these processing techniques as a representation. The techniques investigated thus far include N point reduction; number density reduction; distribution and spectral analyses; and piston, root mean square, and tilt (PRT) analysis. In each case the processed data can be analyzed as a function of a few variables and a simplified pattern results. These simplified patterns can then be compared with one another to form classes and ultimately allow a discrimination process based on these simplified pattern classes.

These techniques are summarized below in table 2 and discussed further in the subsequent subsections.

Technique	**Variables**	**Discriminant**
N Point Reduction	Mean and standard deviation	Relative position of subset points
Number Density Reduction	Number of Points per Cell	Relative matching of image arrays
Distribution(Histogram) Analysis	Characteristic parameters of the distributions	Degree of overlap between distributions
Spectral Analysis	Real & Imaginary Coefficients	Radius Length
PRT Analysis	Piston, RMS and Tilt	Dot product of three vector

Table 2. Summary of Processing Techniques

Two types of classes are discussed in this report. The first is the *object class*. This class is derived from an N point reduction analysis and is based on the magnitude and variance of the data (as discussed in subsection, 2.2.1). Applying this analysis to the data results in one of four classes; class 1 (RV), class 2 (tank), class 3 (PBV) and class 4 (fragment).

The second type of class is the *pattern class*. This class is derived from the other processing techniques and is more a function of the visual appearance of the data. These techniques identify the different patterns existing in the data. They then group the data, according to certain features (discussed later in this section and in subsections 4.3.2 and 4.3.3), into an arbitrary number of pattern classes.

2.2.1 N Point Reduction

In this approach, the original data set is divided into subsets of N points each. While N can be any user selected integer, in this particular analysis, N has been set to 20. For each subset the mean and standard deviation are calculated. It is found that this approach illuminates the general character of the original data set in terms of averages and variability in a fairly simple and effective way.

It should be noted that given the relatively low track rate, the individual points within each subset are essentially independent in the following sense. The low track rate implies a relatively long period of time between pulses. Thus the return due to any given radar pulse will not be influenced by the previous one, since the current induced by the previous pulse has decayed and is no longer a source for radiation. Therefore, any correlation among the points and subsets should only be a function of the object's physical characteristics and trajectory.

The usefulness of this approach is seen when the subset means and standard deviations are graphically displayed, for example, in figure 3 below. Here we can see how the points corresponding to the data sets for objects 2348 and 2609 occupy distinct regions within this plot.

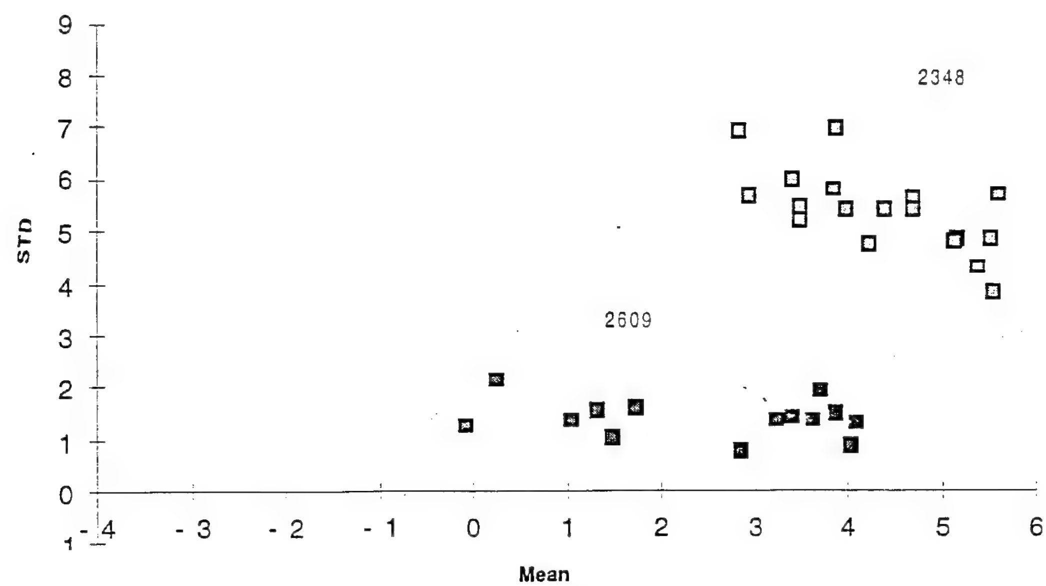

Figure 3. N Point Reduction Plot for Objects 2348 (Tank) and 2609 (RV)

Within this context we have also considered higher moments. Generally the use of higher moments should be used with caution. However in this case we are simply looking for a qualitative picture of the various test sets. The question of interest is then, are the data distributed about the mean in particular ways? If this was so it might indicate a specific scattering shape or a particular rotational motion along the object trajectory. In particular, the third moment, the skewness measures the relative extent of the tail toward either smaller or larger amplitudes. The fourth moment, called the kurtosis, measures the relative flatness of the data distribution. The flatness is measured relative to a normal distribution. The problem with these moments is that it is difficult to decide when a particular value of the skewness or kurtosis is significant. The formula used to calculate the skewness is

$$\mathrm{skew}(x_1 \ldots x_1) = (1/N)\sum_{j=1}^{N}\left[x_j - \bar{x}/\sigma\right]^3$$

where σ is the standard deviation. It is seen that in general any N points drawn from a symmetric distribution will have a non zero value when plugged into the above equation. The same is true for the kurtosis. In this case the formula is given as[1]

$$\mathrm{kurt}(x_1 \ldots x_1) = \left\{(1/N)\sum_{j=1}^{N}\left[x_j - \bar{x}/\sigma\right]^4\right\} - 3$$

As a rule of thumb, for the skewness to be considered significant, its value should be several times greater than $\sqrt{6/N}$. In the case of the kurtosis, the value should be larger than $\sqrt{24/N}$.

2.2.2 Number Density Function

Next, we consider a number density approach. In this case we are using a reduction process where the precise amplitude is not the key discriminant. We divide an RCS Vs time plot for any particular data set into a fixed number of cells. The number of data points falling within each cell is then counted. At this point the cells and their count can be represented as a matrix, with the columns representing the time blocks. One can envision performing a numerical analysis of the matrix. For example, if the data is statistically stationary, then the columns of the matrix will all be parallel and the column space of the matrix will be one dimensional. Performing a singular value decomposition on the matrix will demonstrate this, in that only one non-zero singular value will be

[1] See for example W. H. Press, et al., "Numerical Recipes (FORTRAN)." Cambridge University Press, Cambridge, 1989.

present and hence the actual rank of the matrix is unity. Thus for truly stationary data, the matrix has only one singular value.

If however, each column is markedly different from the rest (i.e., the columns in the matrix form a linearly independent set), then all of the singular values will be non-zero indicating a completely nonstationary data set. The situations in between these extremes can be quantified by the distribution of singular values. As a nonstationary data set can imply a misassociation, a singular value decomposition of the matrix can provide an indication that this is indeed occurring.

Moreover, other more pictorial analyses can certainly be realized. These can take the form of either two dimensional surface maps or three dimensional column plots in which the gray scale or height of the column can represent the numerical value of each cell. This clearly displays the degree of structure inherent within the data pattern and might make a useful adjunct display to an operator. Examples of these are shown below in figures 4 and 5. What is seen in these cases is the degree of structure present in the data. That is, do the data show a relatively even spread as in figure 4, or are the data essentially constrained to only certain cells, such as in figure 5? This degree of structure or organization suggests the possible use of entropy methods to do pattern recognition. That is the pattern discriminant would be the degree of disorder or entropy characterizing the particular data set.

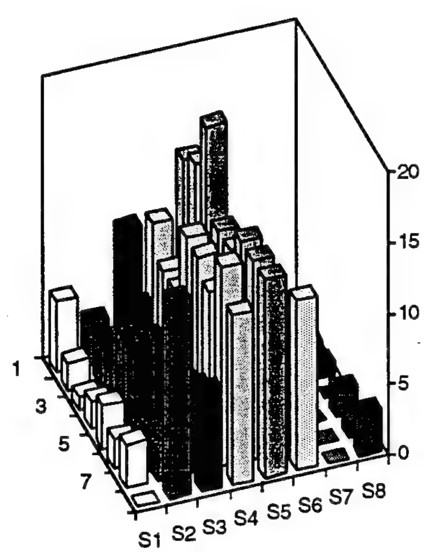

Figure 4. Three Dimensional Column Plot for Object 2348 (Tank)

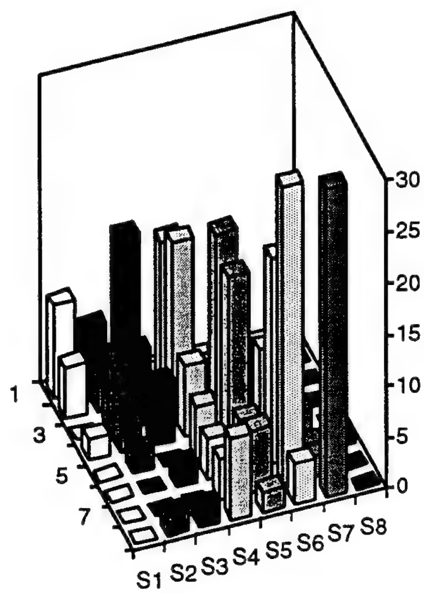

Figure 5. Three Dimensional Column Plot for Object 2609 (RV)

However our primary interest in this approach will be to use it as a way to construct pattern classes, and this will be discussed in section 4.3.2.

2.2.3 Distribution Analysis

The next approach is to consider the entire data set and construct histograms, probability distribution functions (PDFs) and cumulative probability distributions (CPDs) for each set. A histogram is generated simply by dividing the range of RCS values into a series of bins of a given width. The number of data points falling into each bin is recorded. This gives a simple picture of how the RCS values are distributed. The PDF is approximated by a normalized version of the histogram. That is, we take the number of points in each bin of the histogram and divide it by the total number of points in the data set. The normalized histogram gives the relative frequency of occurrence of a particular RCS value in the track file. This is an approximation to the PDF of the RCS data. The CPD is formed by calculating the percentage of RCS amplitudes that are greater than any particular amplitude value within the given range of the data set. In all cases the shapes of these distributions will depend on the character of the original data set.

2.2.4 Spectral Analysis

Another approach also considers the data set as a whole. In this case one applies spectral techniques to the entire RCS time history data set and out of this we get various measures of the frequency content. Among such techniques, one can examine the power spectrum density (PSD) and the auto-correlation function (ACF). It has also been found useful to simply examine the real and imaginary coefficients of the Fourier expansion. The values of the coefficients are then plotted in the complex plane. The relative spread of the coefficients is a measure of the amplitude of the various frequency components, without regard to their order of occurrence.

2.2.5 PRT Analysis

In this analysis one considers a particular data set and constructs a vector whose components are denoted as "piston", "RMS" and "tilt". These colorful terms are borrowed from optical engineering where similar problems in mirror and image construction are often encountered. The term "piston" refers to the average value of the data set and the "tilt" refers to the amount of linear increase of the pattern as a whole as a function of time. "RMS" or root mean square represents the residue after the piston and tilt are removed from the data set.

As this is only a three vector, its usefulness by itself is somewhat limited, since in some respects it represent only slightly more than what is currently done by the radar to do target classification. However, when combined with the distribution analyses, it becomes an effective adjunct tool. This will be described in more detail in section 4.

2.3 Discussion

We have identified a number of processing techniques that might prove useful in classifying or distinguishing various objects as viewed by the radar. There are strengths and weaknesses associated with each approach.

For example, when the spectral techniques are considered, such as PSD and ACF, i.e. those pattern types that have to do with correlation of the data at one point in its history with the data at another time, we are faced with the problem that we do not have any knowledge of the scattering body's motion. The body's motion determines the location in frequency of various spectral peaks in the PSD, and also the location in time at which the data is self correlated. However, the same body rotating about a fixed axis at a slower or a faster rate will give RCS histories with spectra that are peaked at different locations, corresponding to the two different periodicities of the RCS history. That is each

periodicity is determined by the rate of body rotation. Similar ambiguities occur when considering the ACF.

Thus the PSD/ACF techniques can cause identical targets to have different classifications, due to different motions, even when the motion difference is simply a difference in rotation rate. This problem is analogous to the problem of using the RCS histories themselves as pattern vectors. Here again, we do not know the motion of the body from the EWR data and hence we do not know the aspect angle history of the body. If we knew the set of aspect angles (aspect angle history) of a return, then we could go to a library of RCS Vs aspect angle calculations or measurements, and deduce which target gave the return. This is why it is useful to create a representation that is insensitive to the time of occurrence of each RCS sample.

The PDF/histogram representations retain the data itself and discard *when and in what order it occurred.* The histogram simply counts up how many times in the data set the RCS values fell into each RCS bin. If one normalizes this to form the relative frequency of RCS occurrence, we then obtain an estimate of the PDF. Similarly, plotting the Fourier coefficients in a complex plane with no regard to their spectral location is a means of preserving information in the data and ignoring the difficulty encountered when trying to interpret why certain coefficients are at certain spectral locations, when no information about motion is available. This procedure is a sort of histogram taken on the complex Fourier transform of the data. One can also take a direct histogram on the PSD as a pattern class.

Another processing technique that is useful when the RCS exhibits non-stationary behavior is the PRT representation. In this case, we calculate the DC level or "piston" dependence of the RCS history, the linear rate of increase (slope or "tilt") of the data, and the residual RMS left over after the piston and tilt terms are removed. In addition, the peak and minimum level of RCS may be used in this pattern vector.

In fact a body that is of the order of a wavelength or less, (such as RVs and fragments) will have few or no minima as a function of aspect angle. However, a larger body such as a tank will have a number of minima and peaks. Thus having this data as part of our vector can improve its performance as a discriminant.

We also have an option to remove the tilt from the data and form the histogram/PDF pattern vector with the remaining data. We can then use the two pattern vector types (PRT and PDF) together in a weighted linear combination to do the classification.

Section 3
Initial Data Survey

3.1 Objective

In this section we review the results of applying the data processing techniques discussed in section 2 (with the exception of the PRT analysis, which was developed after the survey was completed) to a large number of data sets.

Our initial discussions presented in PR I were based on the examination of a limited number of data sets. The next step was to enlarge our data base. The initial results suggested that there were a small number of basic patterns in the data and that the various data processing techniques allowed a way in which to enhance various aspects of these patterns.

Note that much of this material was discussed in PR II. It is presented again here with some amplification to demonstrate the utility of applying the processing techniques using specific data sets as examples. The results of this survey provided the motivation to develop a classifier based on pattern recognition techniques.

Before presenting our results, we first review the data analysis procedure.

3.2 Data Set Selection

Conceptually one can divide the RCS data base into three groups (i.e., groups I, II and III) corresponding to the date on which they were collected by the radar. Each set has the following information as discussed in section 2.1: the track file ID number; the time; the RCS and X, Y and Z position of the object in radar face coordinates. Thus the position and RCS are both given as functions of time. Furthermore, the range, azimuth and elevation of the object can also easily be obtained through simple transformations involving the radar face coordinates.

At this point, we have completed our survey through groups I and II. In fact all of the data sets within group I have been visually examined, amounting to nearly 60 sets. However, a significant fraction of these are either deficient in the number of sample points or do not span a sufficient length of time, i.e., 100 seconds or more, to be consistent with the global nature of our approach. As a result only 28 data sets from group I were appropriate for detailed analysis. For group II, we were more selective right from the start, in that we generally demanded a minimum track data rate of 0.25 Hz and a period of observation of 100 seconds or so before we would even visually examine the data set.

With these selection criteria applied, the number of processed data sets from group II totaled 24.

The data in group III was set aside for use in testing the prototype classifier. This will be discussed more in section 5.

3.3 Description of MATLAB

As a tool to generate and assess various data representations, we used MATLAB, a numeric computation and visualization software package. There are a number of advantages to using MATLAB. First of all it offers a possible way of emulating the entire pattern recognition system. It has a powerful Graphical User Interface (GUI) capability that allows for the generation of menu driven processes and various types of controls (push buttons, sliders, etc.) which ultimately control an excellent graphics package. Thus one could simulate the entire pattern recognition process within MATLAB. This simulation could be run in an automatic mode or with an observer in the loop, since all intermediate data and displays can be brought to the screen. The potential benefit to further algorithm development and debugging phases is obvious.

However at a more immediate level, MATLAB offered an easy way to visually survey a large amount of data. Data sets from each group were loaded into corresponding "work spaces" in the MATLAB environment. Each file was then run against a script or program consisting of MATLAB commands which worked through the various data processing techniques (plus additional related plots) and outputted the results to the screen. The figures shown in subsection 3.4 illustrate the utility of both the data processing techniques and of MATLAB.

3.4 Summary of Data Survey

In this section, we will concentrate principally on objects we believe to be RVs and tanks. This classification was determined by letting MATLAB do an N reduction on the data sets as part of its script.

Some of the results of the MATLAB survey are presented below.

Figure 6 presents the basic picture of the data, i.e., the RCS amplitude as a function of time, (RCS time history) given in seconds relative to the start time of the event. We feel that this represents the typical pattern for a tank or class 2 object.

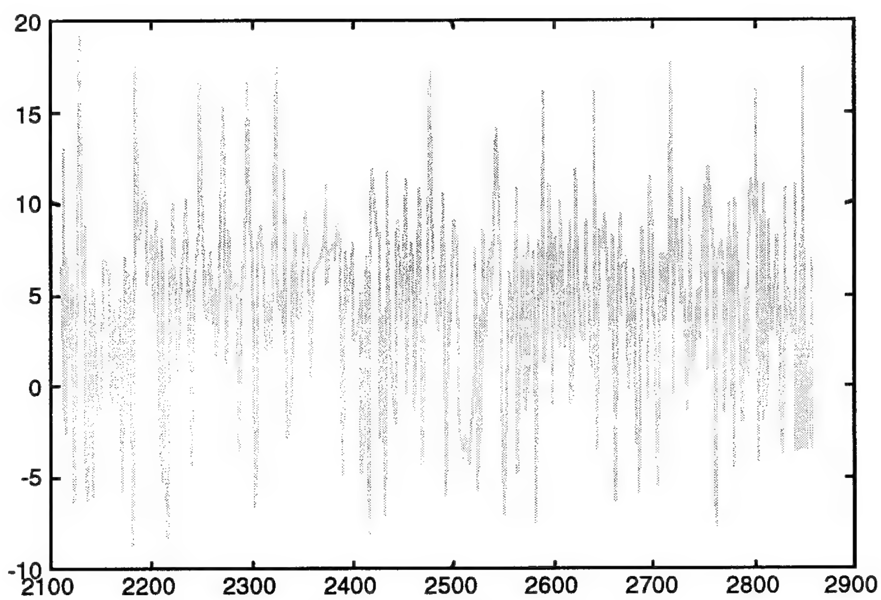

Figure 6. RCS Vs Time (Object 2348 Tank)

The time is given by the horizontal axis, while the dependent variable, in this case the RCS amplitude, is given by the vertical axis. The pattern is characterized by a large variance and a relatively high mean value.

We now consider a probable RV data set. This particular data set was discussed briefly in PR I. The RCS time history is shown in figure 7 (next page).

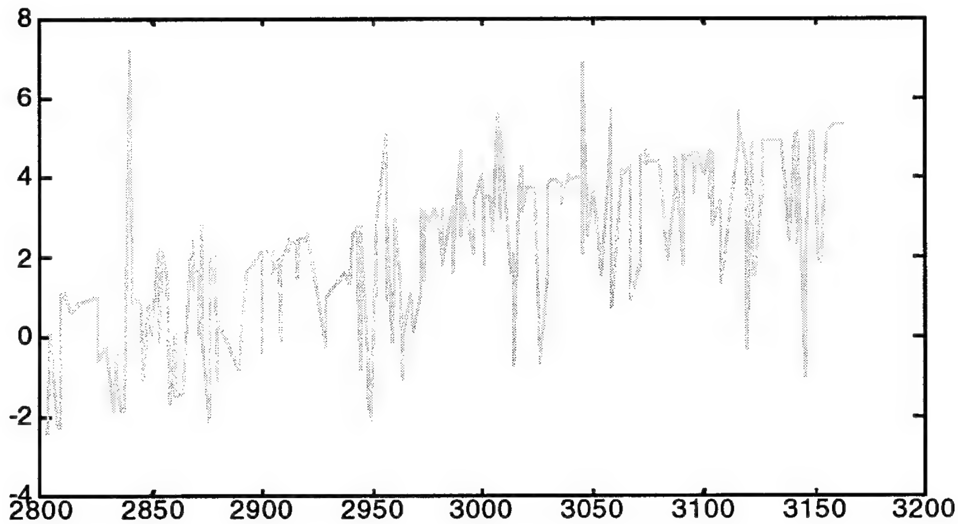

Figure 7. RCS Vs Time (Object 2609, RV)

The next step in the MATLAB script produces a histogram, a probability density function (PDF) and a cumulative probability distribution (CPD) of the data. The width of the bins of the histogram were set at 0.5 dB. The PDF is essentially a smoothed version of the histogram with the entry in each bin normalized by the total number of points. Thus the total area under the PDF curve is equal to unity. These plots are outputted together and shown in figure 8 (next page).

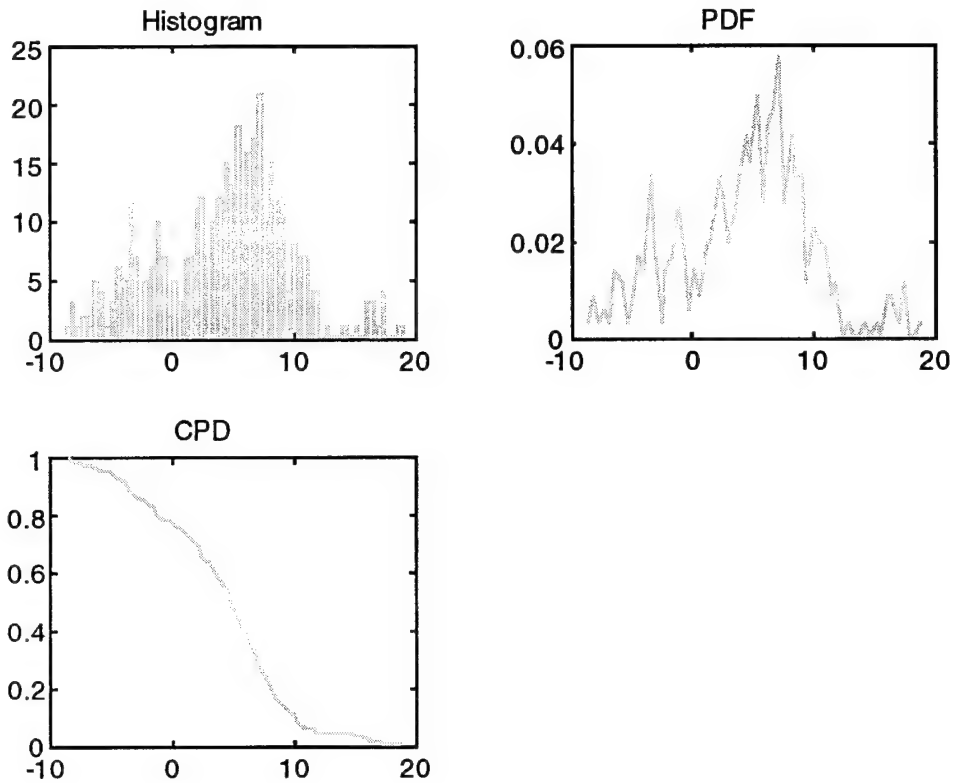

Figure 8. Histogram, PDF and CPD (Object 2348, Tank)

The characteristic of the tank signature is seen in the relative spread of the histogram and CPD. In fact we see a significant amount of data spread over about 20 dB. The CPD gives directly one measure of the center of the distribution, i.e., the median. In this case we note that the median is about 5 or 6 dB. This should be compared to the same plots for the RV data set shown in figure 9 (next page).

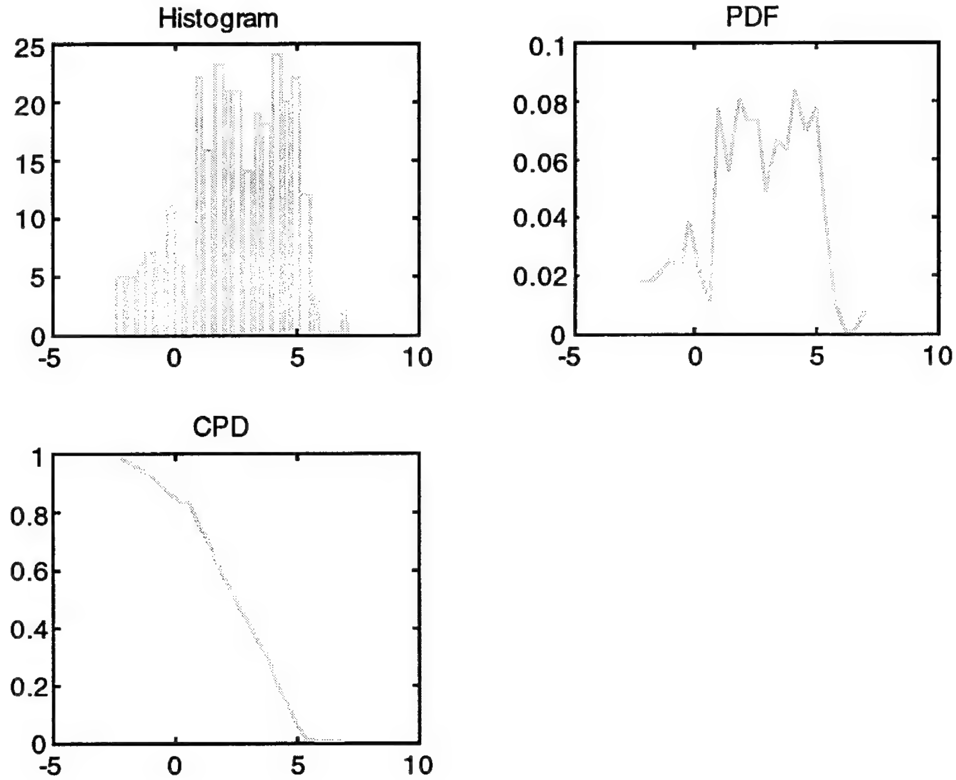

Figure 9. Histogram, PDF and CPD (Object 2609, RV)

The histogram in figure 9 indicates a narrow variance with most of the values falling between 0 and 5 dB. This is to be compared to a 20 dB range for the tank example.

One should also note the asymmetry in the histogram due to the large number entries into high amplitude bins. This is a reflection of the systematic increase in the RCS signature over time, which may be due to a slow variation in the aspect angle as discussed in PR I.

Since the PDF is essentially the normalized version of the histogram, it displays the same basic shape as the histogram. However, it may present an easier pattern for an algorithm to identify. The CPD represents another way of indicating the basic characteristics of the data. While from a visual point of view, the information may be more readily apparent in the histogram or PDF, we have actually used all three representations in our final classifier.

MATLAB then produced the Fourier coefficients (at two scales) which are given below in figure 10.

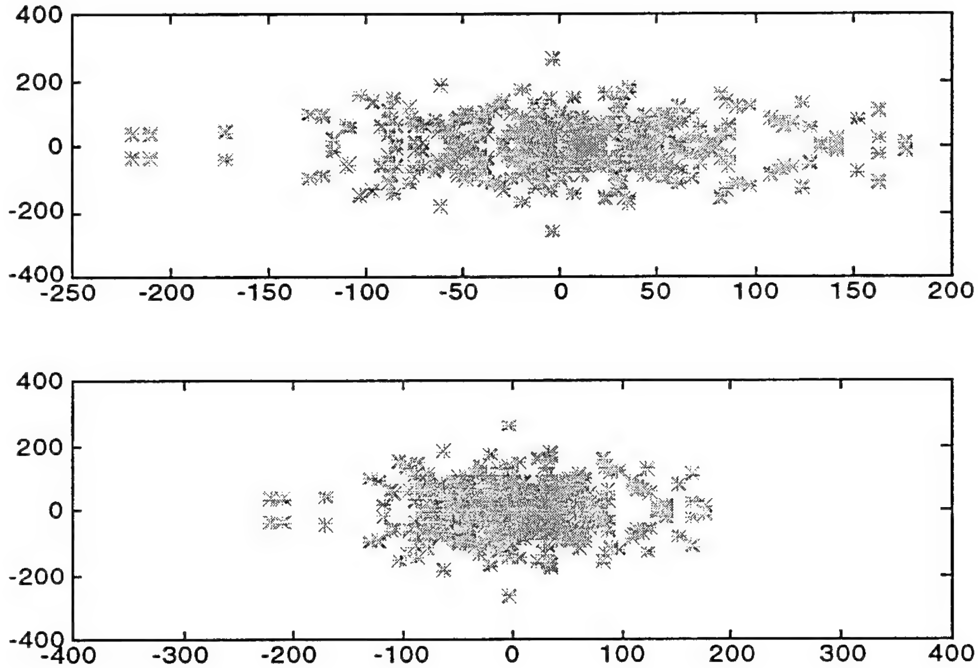

Figure 10. Fourier Coefficients (Object 2348, Tank)

In this case we imagine drawing a radius line from the center of the complex plane out to a point which would enclose a majority of the coefficients. Assuming for the moment that we are dealing with dimensionless numbers, the required length of the line would lie somewhere between 100 and 150 depending on how we defined a majority. This should be compared to the RV example shown on the next page in figure 11.

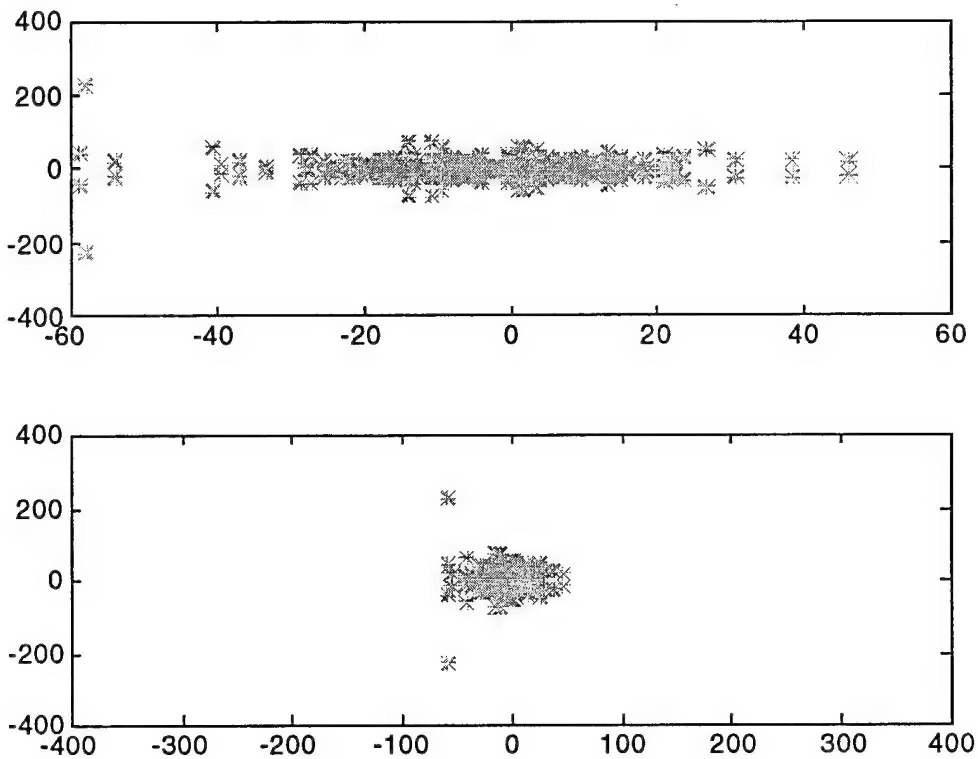

Figure 11. Fourier Coefficients (Object 2609, RV)

Next in the MATLAB script is the power or the amplitude squared of the data (given by the vertical axis) as a function of both the frequency (in Hz) and the period (in seconds), given by the horizontal axes. For the latter, it is presented at two different scales. This is shown on the next page in figure 12 for the tank object.

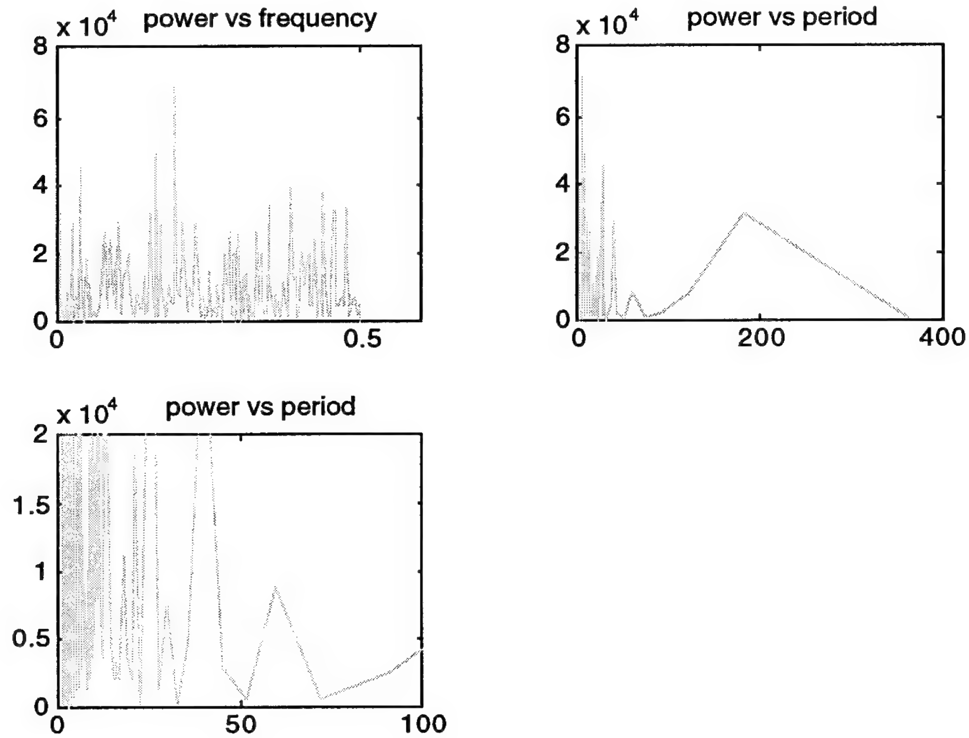

Figure 12. Spectral Data (Object 2348, Tank)

Of most interest is the power Vs frequency plot given in the upper left hand corner of the figure. Here we see that the power is distributed over the frequency range in a fairly even manner. The corresponding RV example is presented in figure 13 on the next page.

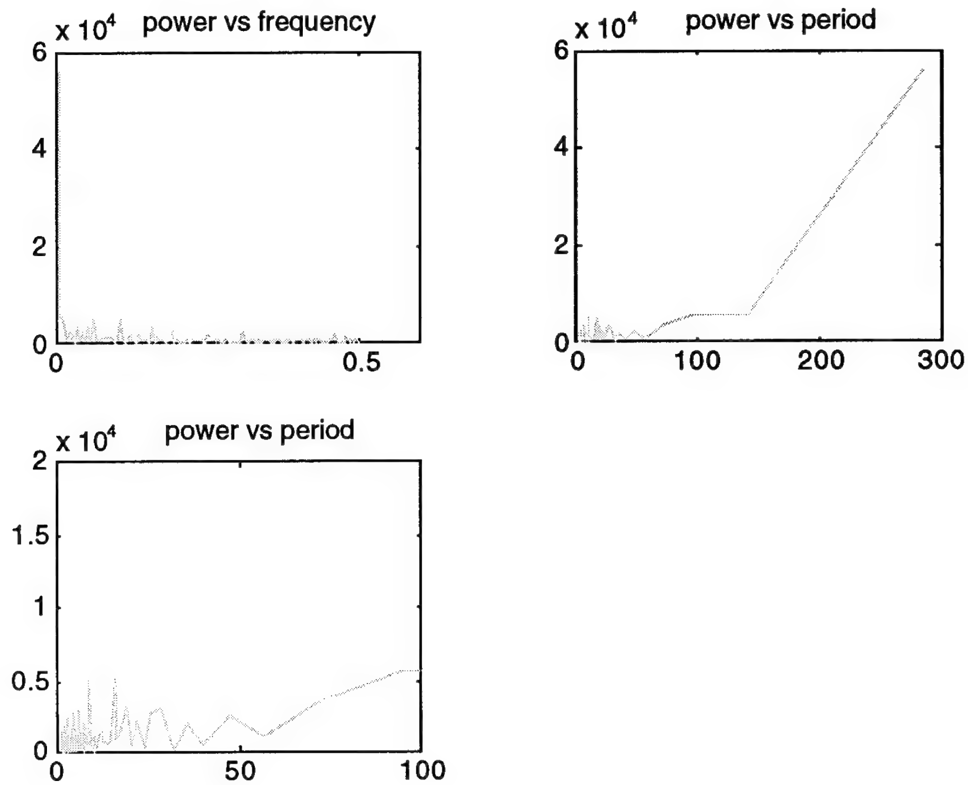

Figure 13. Spectral Data (Object 2609, RV)

Figure 13 presents the same spectral plots as shown previously for the tank. Again we are considering the amplitude squared (power) as a function of the frequency and as a function of the period. In the later case the data is presented at two scales.

If we first consider the frequency response, we note that most of the power is taken up by a near zero Hz line. This is due to the strong DC component of the data. The straight line feature in the power Vs period plot reflects the linear increase of the RCS over time or aspect angle.

A subtle point should be made here. Since we are using PAVE PAWS track file data, the RCS is actually in a dB scale. If we were concerned the exact frequency content, then the RCS should be converted to amplitude before a spectral analysis was made. However, our interest is in the patterns and their features are often best illustrated in a dB scale.

Next, in figure 14 below, we show the distribution of the 20 point means and standard deviations. The mean is measured along the horizontal axis and the standard deviation is given along the vertical axis.

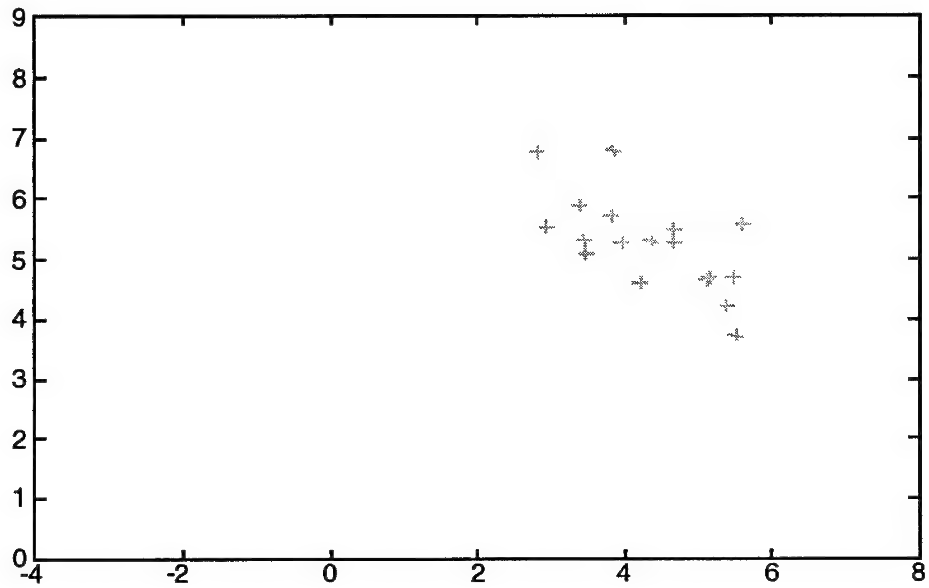

Figure 14. 20 Point Means Vs 20 Point Standard Deviations (Object 2348, Tank)

We note that a majority of subset points lie in a region that has been designated as class 2, i.e., a tank region.

The points in figure 14 are actually functions of time and thus we could easily present a three dimensional plot showing how the subset means and standard deviations change with time. However it is actually clearer to take two dimensional projections through that plot and show the subset means and standard deviations each as explicit functions of time.

This is done in figure 15 where time is measured along the horizontal axes.

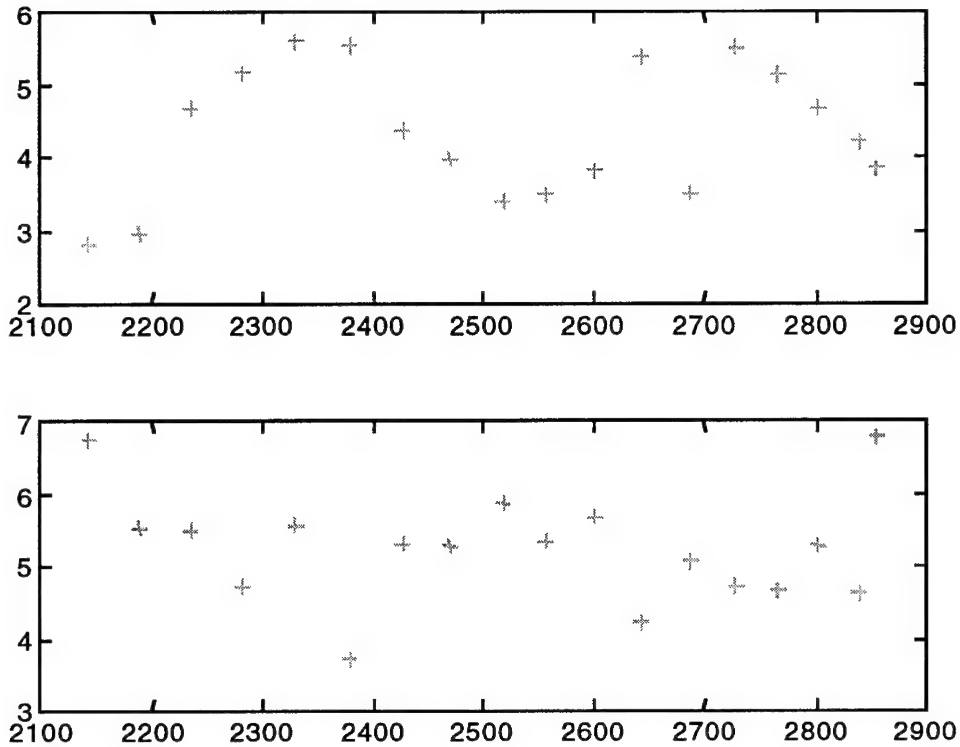

Figure 15. 20 Point Means and 20 Point Standard Deviations Vs Time (Object 2348, Tank)

The values of the means and standard deviations are generally large, although there is some fluctuation with time. We also note the almost periodic variation in the subset means.

We then compare these plots to the corresponding set for the RV object found on the next page.

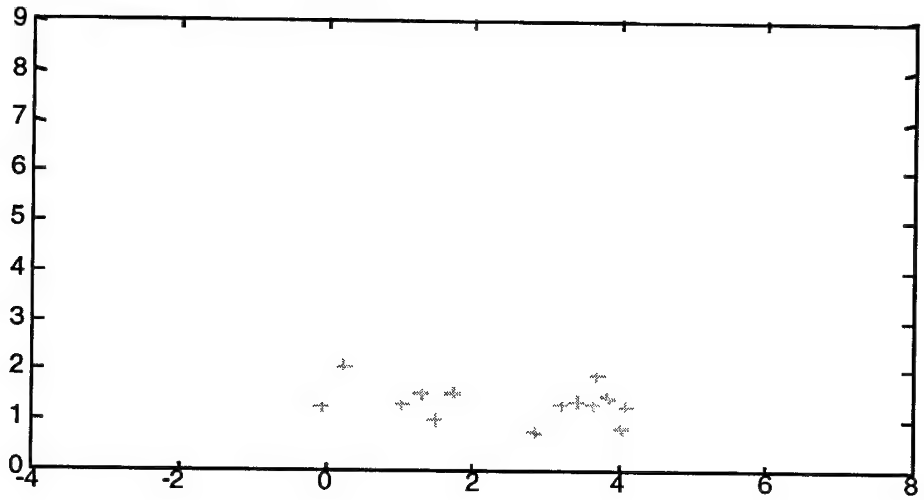

Figure 16. 20 Point Means Vs 20 Point Standard Deviations (Object 2609)

It is seen that the characteristics of the RCS plot force the subset points in figure 16 to lie in a region totally different from what was seen in figure 15 for the tank example. This is not surprising given that the two RCS signatures (figures 6 and 7) are quite different. However, recall that the aim is to develop simple representations to allow a computer program to make the distinctions.

Finally, figure 17 (next page) shows the time variation of the means and standard deviations for object 2609. Here we see that the increase in the RCS is reflected in the increase of the subset means as a function of time. However the variance or standard deviation remains small and relatively constant over the data collection time.

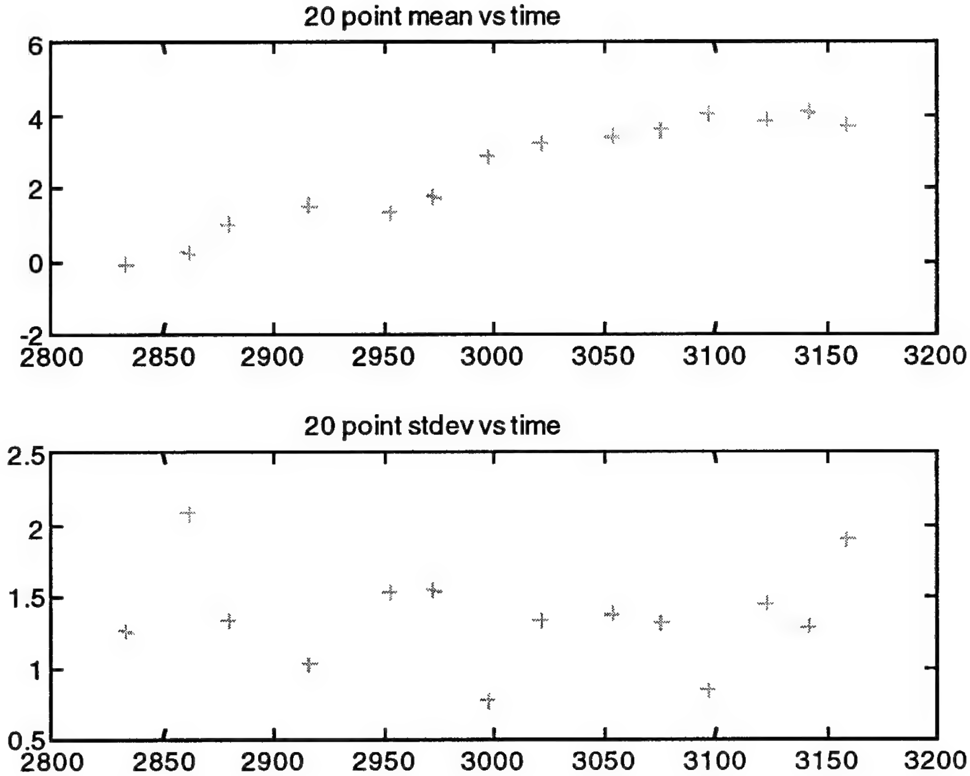

Figure 17. 20 Point Means and 20 Point Standard Deviations Vs Time (Object 2609, RV)

During our analyses of data groups I and II, we have discovered a number of cases that have RCS patterns that are similar to object 2609. Examples of these are shown in figures 18 and 19 (next page).

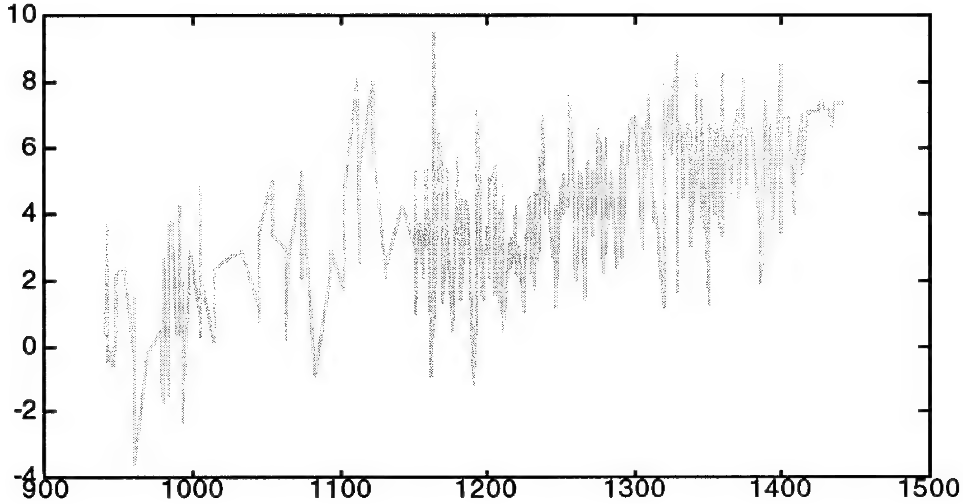

Figure 18. RCS Vs Time (Object 6221, RV)

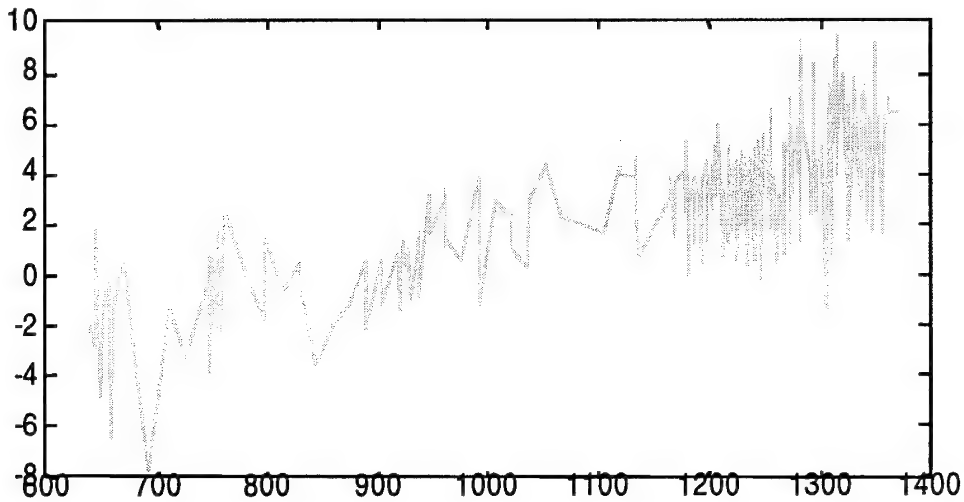

Figure 19. RCS Vs Time (Object 6193, RV)

At this point, since the histogram and spectral plots for these two cases are quite similar to that seen for object 2609, it is more illuminating to consider an object that is classified as a class 1 object but whose RCS pattern appears different.

For this case, we consider object 1234 whose RCS time history is given in figure 20 (next page).

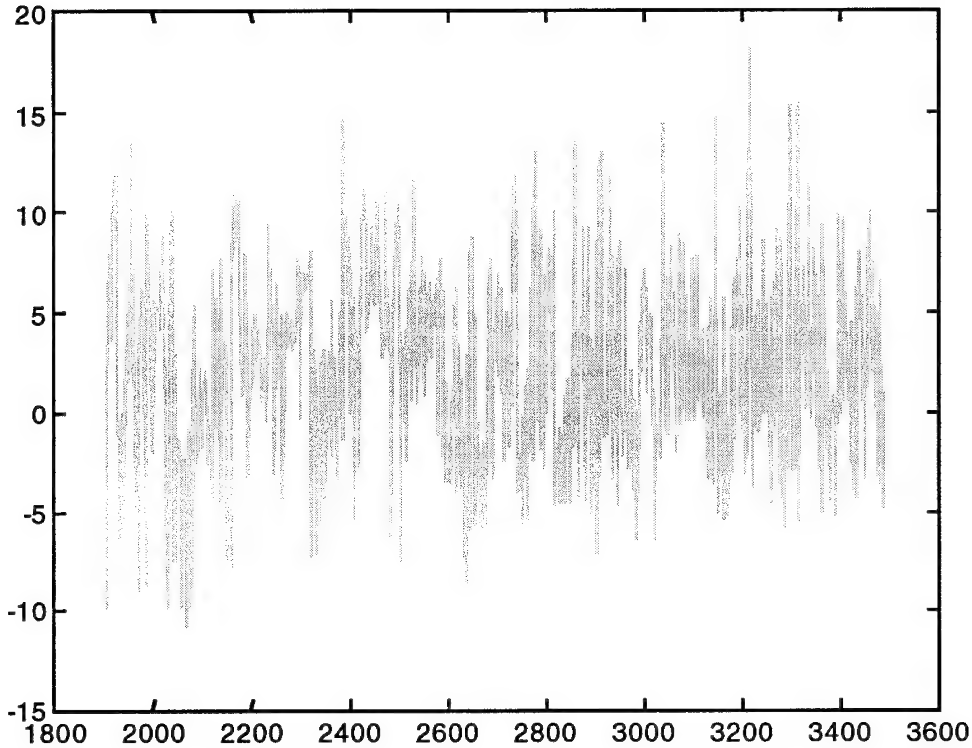

Figure 20. RCS Vs Time (Object 1234)

While our primitive classifier assigned this data set to class 1, it is obvious that the RCS signature is different from what we saw for object 2609 (see figure 7). The question is whether this difference is reflected in the histogram and spectral plots.

The histogram data for this object is reproduced in figure 21.

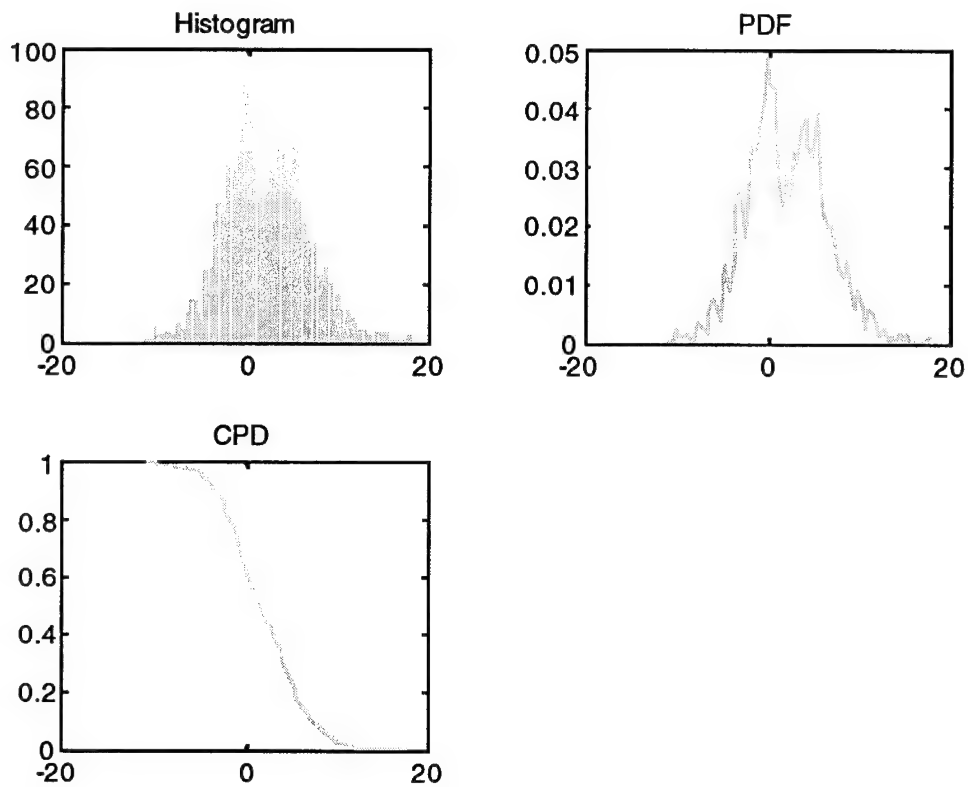

Figure 21. Histogram and Related Plots (Object 1234)

In this case the shape of the histogram is quite different from what was seen for object 2609 (see figure 9). In fact the above histogram appears almost bi-modal. While this may be accidental, it could also indicate the result of receiving returns from two targets within the same range resolution cell of the radar. This problem will be discussed in more detail below. What is significant is the fact that the range of values of the histogram for object 1234 is nearly four times the range seen in the histogram for object 2609. These differences are also seen in the PDF and CPD.

The differences in the RCS time histories are also seen in the plots below for the Fourier coefficients.

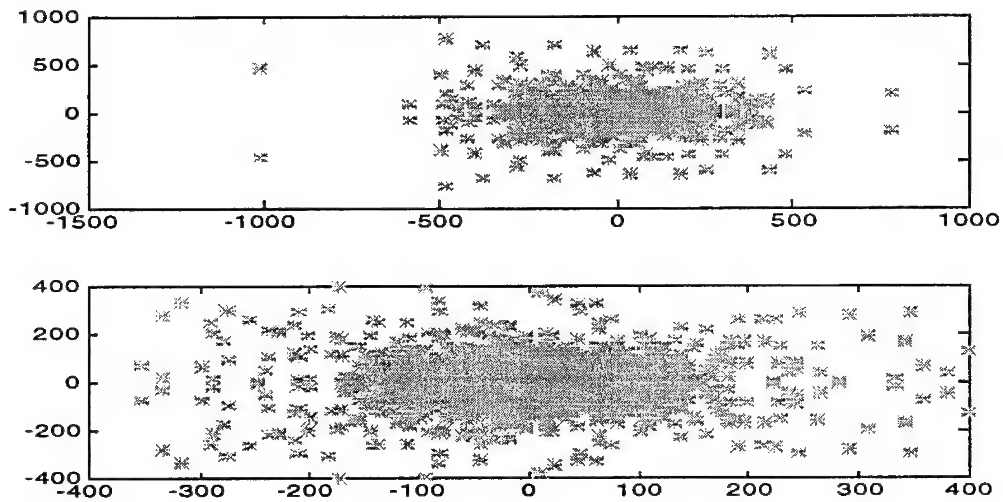

Figure 22. Fourier Coefficients (Object 1234)

The spread along both the real and imaginary axis is much more pronounced than for object 2609 (see figure 11). It is even more pronounced than what was seen in the tank example (figure 10).

Now consider figure 23 which displays the spectral data.

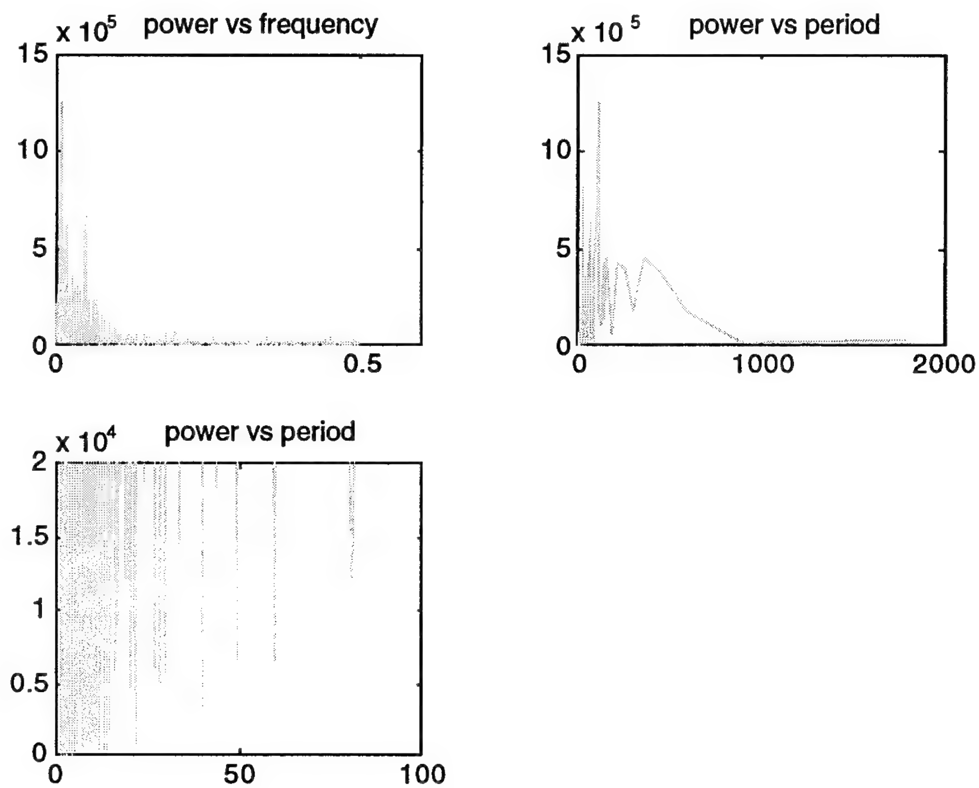

Figure 23. Spectral Data (Object 1234)

There are a couple of things to note. Although it is not obvious due to the way the y axis is scaled, the power in the frequency domain is more than an order of magnitude greater than what was seen for object 2609. Moreover the frequency content is different in that a significant portion of the power is distributed over frequencies beyond 0 Hz.

We also note that the linear characteristic in the power Vs period for long periods is absent for this example. This is to be expected, since the RCS time history (figure 20) displays no systematic increase in RCS as a function of time.

We see that based on a simple classifying scheme, objects with different RCS patterns can both be assigned to the same class. But this is to be expected, since the classifier is sensitive to only the grosser aspects of the pattern, that is to the 20 point means and

standard deviations. The good news is that we appear to have tools (histograms and spectral representations) which are sensitive to other aspects of the pattern and thus the two data sets are distinguishable. This is not to say that one pattern represents an RV and the other does not. It could well be that both are RVs and the differences in patterns are due to differences in shape, size or body motion relative to the radar line of sight (i.e., tumbling). But we can distinguish the patterns, and when truth data is available, we can begin to assign these patterns to particular objects.[2]

There are of course some issues which need to addressed. First the reader may have noticed that in figures 18 and 19, while the patterns of the RCS were quite similar, the amplitude scales are somewhat different. This could indicate the presence of larger objects that share some of the characteristics associated normally with an RV. The best solution to this difficulty is to establish a threshold based on a reasonable estimate of the RV's RCS. This can be obtained through modeling exercises or from truth data.

A related difficulty is illustrated by figure 24.

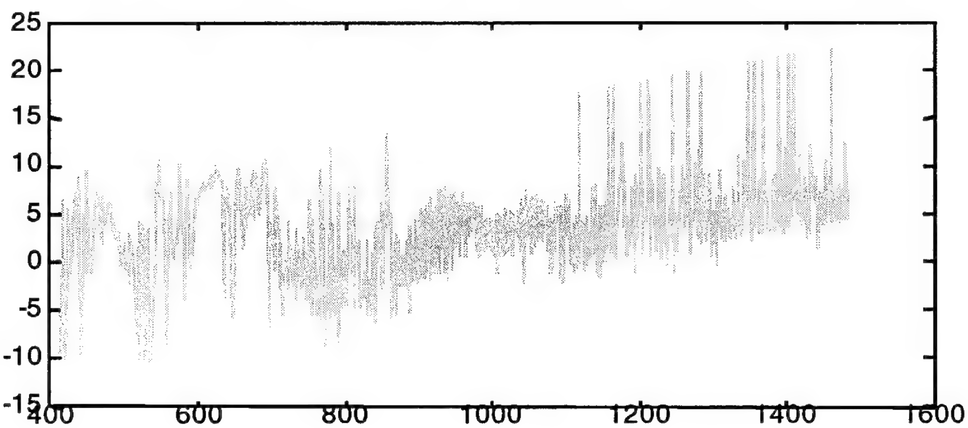

Figure 24. RCS Vs Time (Object 6153, RV)

In this case there is a general increase in the RCS starting at around a time equal to 800 seconds. However, up until that point the pattern is quite different. The problem may be one in which the radar is placing returns from more than one object into the same track file. This problem is termed a misassociation and is not uncommon with these types of radars. However from our point of view, the interesting point is that our routines consider this to be a object class 1 object whether there is a misassociation or not. This is the correct assignment, initially at least, since much of the data exhibits a small mean and

[2] Truth data has been requested from the Navy. We plan to to use this data to verify our results in a future phase of this research project.

narrow variance. The fact that the first 400 seconds or so show a different pattern is not important. The fact that there are a small number of high spiky returns at late times is also not important. What is important is that we do not classify a data set like this as something other than an RV during the first cut.

In any case we should be able to develop algorithms to detect misassociation or at least a discontinuous change in RCS pattern. The late time spikes could be detected by using a high pass filter, but are actually already indicated in the histogram of the data as shown below in figure 25.

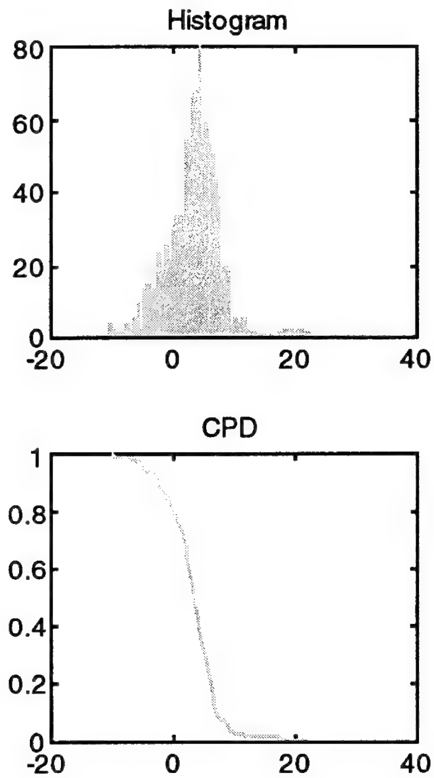

Figure 25. Histogram and Related Plots (Object 6153, RV)

Here the spikes show up in the histogram as a small clustering around bin 20 and are well separated from the main part of the histogram. The spread of the histogram is more than what one might expect for a class 1 or RV object and may be due to the pattern of the RCS returns during the first 400 seconds. In cases such as these, the classifier should probably divide the data set into thirds or quarters and test for a linear increase in the power Vs period plot, which indicates the presence of a systematically increasing amplitude of the return. We simulated this by excising the data from the initial point out to time 800 and then redoing the histogram.

The results are shown below in figure 26.

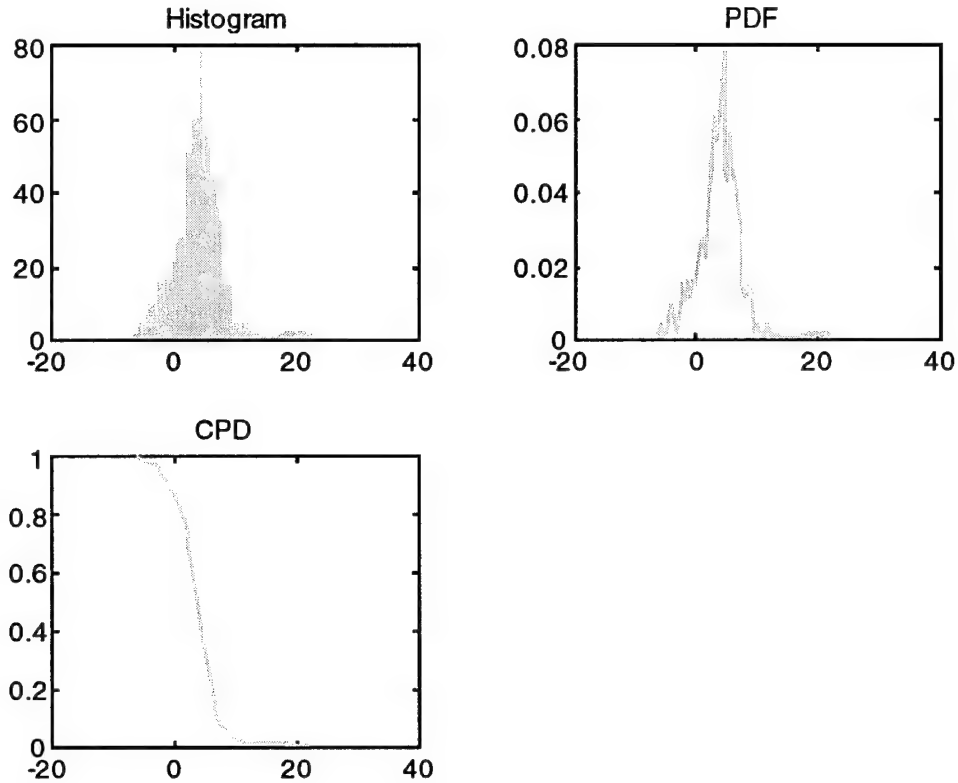

Figure 26. Histogram and Related Plots with the Time Period of 400 to 800 Seconds Removed (Object 6153, RV)

In this case we see a slight difference in the width of the histogram, and we suspect that this data set represents a borderline case. In fact our initial classifier (based on the N point reduction process) is not insensitive to this and does give the percentage of subset means and standard deviations which were actually placed in class 1. For the case of object 1153, we noted that only about 50% of these were actually assigned to class 1. Likewise for the data set on object 1234, we discovered that only about 55% were assigned to class 1. On the other hand for the cases of objects 2609, 1193 and 1221, the percentage of mean and standard deviations assigned to class 1 was 88% or higher. So it may well be that our initial classifier can indicate which data sets will require more work. That is which sets may require high pass filtering, searches for 0 Hz lines or discontinuous pattern change identification before a plausible classification can be made.

One final issue must be discussed and this concerns the use of the spectral methods, i.e., the Fourier coefficients, the power Vs frequency plots and so on. The basic process of taking a discrete Fourier transform assumes the time interval between samples is uniform. For a great number of data sets this is simply not the case, even in a rough sense. The following two figures show the RCS Vs time data and the frequency at which the data was collected for objects 6153 and 1229 (figures 27 and 28).

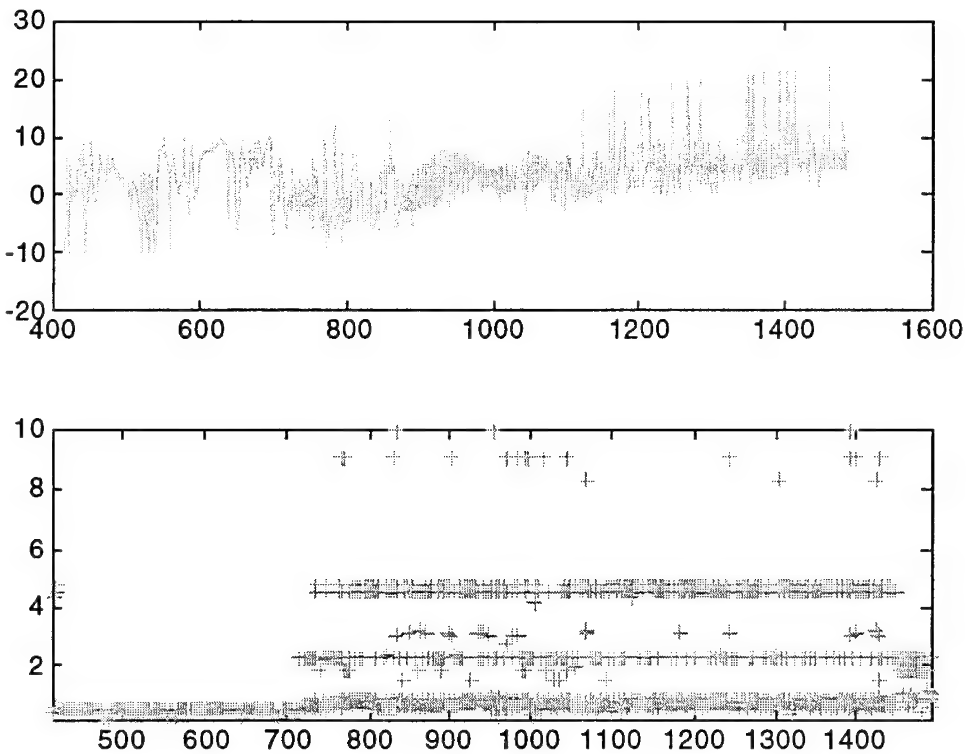

Figure 27. RCS Vs Time and Data Frequency Vs Time (Object 6153, RV)

The first plot gives the RCS time history. The second plot shows the data collection or recording frequency. Thus a value of 10 on the vertical axis indicates a data rate of 10 Hz or a time interval between data points of a 1/10th of a second. At the other extreme, a data frequency of 1/2 Hz means that the time between samples is two seconds. Depending on the physics of the situation, one data rate or another might be preferable. However if the data intervals are constantly changing, then the Discrete Fourier transform (DFT) does not approximate the actual Fourier Transform very well. Unfortunately, this is precisely what is seen in the above data set, i.e., a data interval which changes sporadically with time.

The situation is not completely negative. For example there are some data sets in which the data rates are fairly constant, as shown in figure 28.

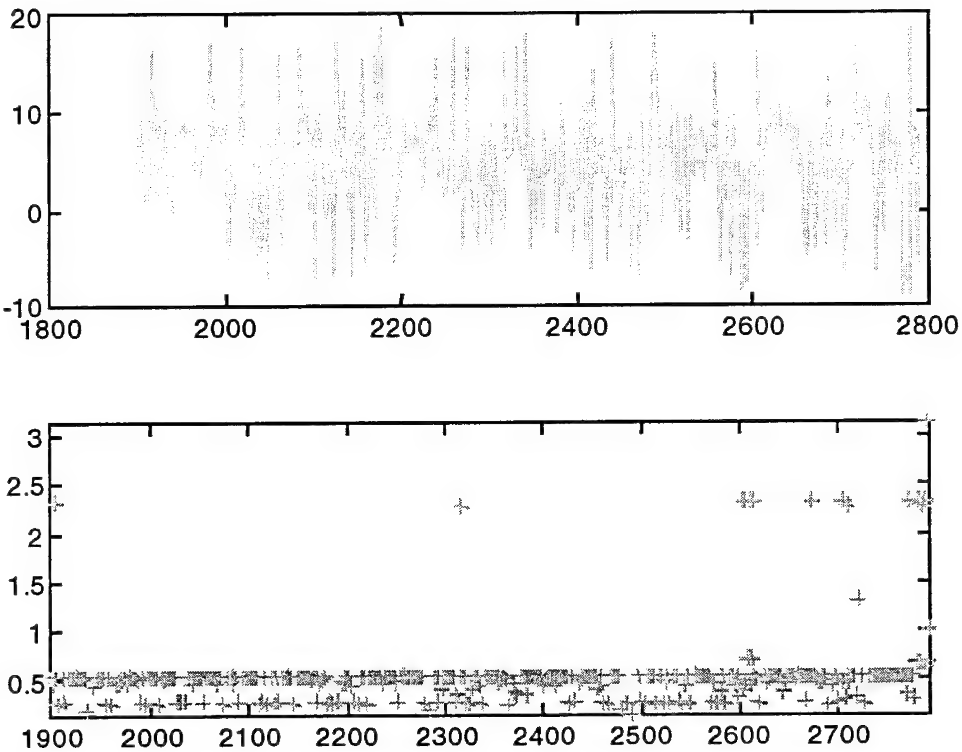

Figure 28. RCS Vs Time and Data Frequency Vs Time (Object 1229, Tank)

It also should be recognized that the information that we are generally seeking from the spectral methods is not of a detailed or quantitative nature. For example, we find that for many of the class 1 objects, most of the power is concentrated at the zero or near zero Hz region. This is a result of a systematic change of the RCS over time and is insensitive to the fluctuating data rate. In addition, we find that for the class 2 or tank-like objects, the power tends to be evenly distributed across many frequencies. This is an expected result, because of the essentially random tumbling motion of the tank. It is also a meaningful result because, in general, the data rates for tanks tend to be uniform as in the example shown above.

In any case, for the spectral methods used in our prototype classifier, we do not use a Fast Fourier Transform (FFT) as MATLAB does. Instead an actual integration is performed using a variable integration interval to compensate for the non-uniform data rates.

3.5 Major Conclusions

There are two major conclusions which result from the analysis presented in this section. First we have seen, by using MATLAB to systematically examine the RCS data base, that there are a relatively small number of distinct patterns that are repeated throughout the data base. Second, we have tools, i.e., processing techniques which are sensitive to the various aspects of those patterns.

This indicates that there is merit in developing a classifier that incorporates these techniques to identify patterns within the RCS data and ultimately use this information to determine the object classification. In the next section, we present a prototype of such a classifier.

Section 4
Prototype Classifier Description

4.1 Architecture

Our prototype classifier program is denoted as Poet, and includes track file reads, subroutine calls and a final classification estimate. A chart displaying the Poet architecture is shown below in figure 29.

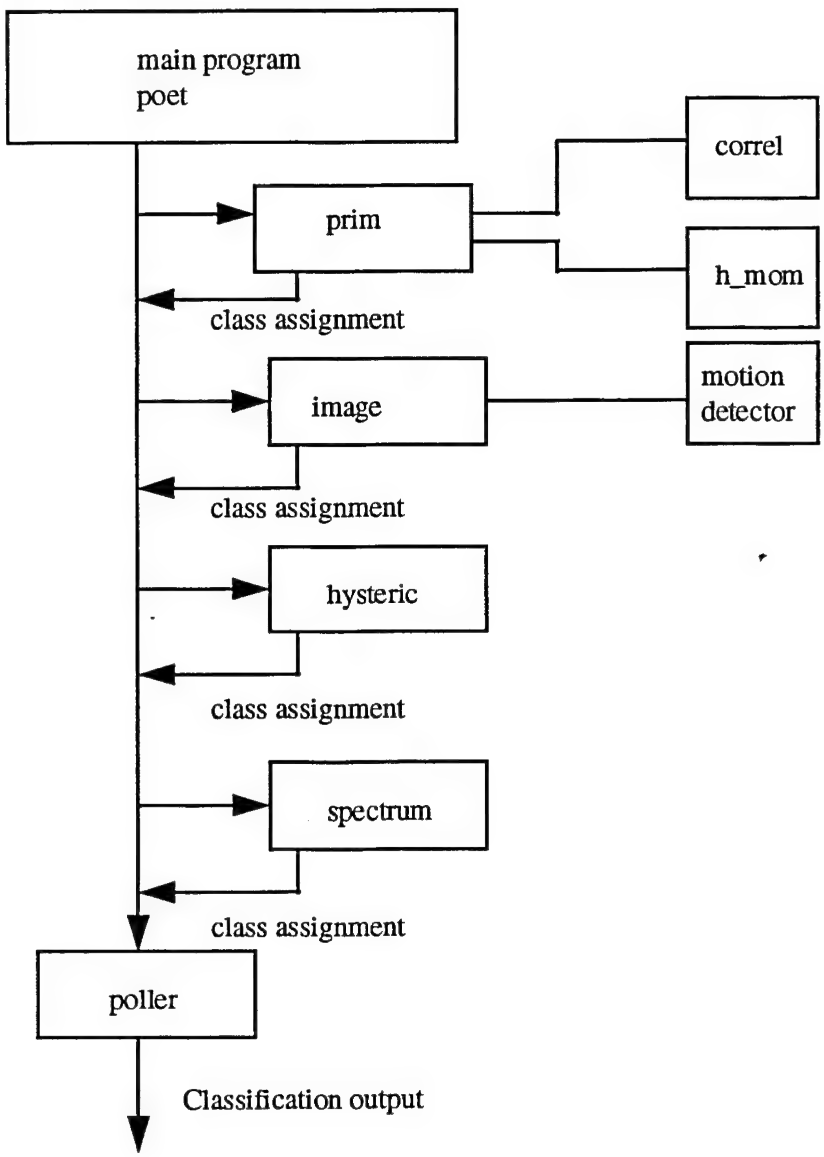

Figure 29. Poet Architecture

All of the basic modules shown in figure 29 are in place with the exception of the motion detector and poller subroutines. The basic thrust of this collection of routines is to assign a particular RCS data file to a given class, based on both long term and small scale patterns seen in the RCS data. Poet calls various subroutines which individually assign the data to a class, based on selection rules that are indicative to the data processing technique incorporated in the particular subroutine.

The program is designed to operate in one of two modes. The first is the so called clustering mode. In this case Poet accepts a large number of data sets, where large implies a range of between 40 and 50. The objective in this case is to use the various subroutines to find a natural grouping of the data sets in terms of their patterns. First, the Prim subroutine divides the data into four *object classes* based on the means and standard deviations of the 20 point subsets. Then the following major subroutines, Image and Hysteric attempt to group data sets with like patterns into a number of *pattern classes*. The subroutine Spectrum currently only attempts to verify the class 1 assignments originating from the Prim routine.

The end result is that all of the data sets are placed into one of a number of pattern classes and into one of the four object classes (tank, RV, PBV or fragment). Thus one can get a feeling for how many different types of patterns one might find within an object class. That is, might RVs, for example, exhibit more than one type of pattern? The clustering mode will be discussed in more detail in section 5.

The program will also operate in a classifying mode. Now the input is a single data set that we want to classify. In this case, the data again passes through all of the various subroutines. However, now the subroutines Image and Hysteric attempt to match the data set to an existing pattern class. These routines have "learned" about pattern classes from running in the clustering mode.

The Poller subroutine (which we will develop later in this research project) will attempt to mediate any disagreements that might arise from the other subroutines. This mode will also be discussed further in section 5.

4.2 Description of Poet

Beyond providing program control and structure, Poet serves as an interface between the LRID track files and the subroutines themselves. Poet searches through the track file and flags those track file identification numbers (IDs) that represent data sets satisfying user specified criteria, such as minimum track length, minimum number of data points and minimum track update rate. Then the RCS, time and track IDs are collected for each of the selected track files and stored in a 3-dimensional array. Currently Poet is capable of storing the RCS and time data for as many as 50 track files.

Poet then passes this array to the various subroutines in the program. This allows the LRID file to be accessed only once, as would occur in an actual operating situation.

4.3 Descriptions of Subroutines

4.3.1 Prim

This subroutine accepts the data set array from Poet and divides each data set into a collection of 20 point subsets. The mean and standard deviation of each subset is calculated. Prim can assign the subset to one of four object classes depending on the value of the mean and standard deviation of the subset. The thresholds or boundaries for each of these classes are currently hard wired into the code and are essentially based on our present understanding of the physics of the scattering processes.

Prim then assigns the entire data set to the object class that contains the majority of the data subsets. The subroutine outputs this information to a file along with the percentage of subsets that actually fell into the selected object class. Thus one gets some indication of how well the data set fits into the given classification.

Prim also makes calls to two other subroutines, Correl and H_mom. The first one, Correl, divides the data set into ten point subsets and calculates the mean and standard deviation of each subset. It then estimates the degree of correlation between the subset means and standard deviations. The subroutine then outputs a correlation coefficient.

H_mom is a subroutine that calculates the third and fourth moments of the entire data set. The subroutine then outputs estimated values of these moments, i.e., the skewness and kurtosis for the given data set.

Since the conditions or rules for class assignment are coded into Prim, this subroutine functions in exactly the same way for both the clustering and classifying modes. It is possible however to make this routine more adaptable by allowing feedback from the other subroutines in Poet.

4.3.2 Image

Image is a subroutine based on the number density technique discussed in section 2. Like the subroutine Prim, this subroutine accepts the data array from Poet. The basic idea is to, conceptually at least, construct a 2-dimensional amplitude-time space and then divide that space into a fixed number of cells. Essentially one can imagine taking an RCS time history plot as shown, for example, in figures 1 or 2 (see page 14) and dividing it into large number of small squares and then counting the number of points within each square or cell. Image does this and then reads the first data set from the array into this space, placing each data point into the proper cell. Once this is done Image goes back and

normalizes the cell entries in each column by the total number of points in the given column. The end result is a two dimensional image array whose entries range from 0 to 1.

If the program is operating in the clustering mode, this subroutine will construct an image array for each data set in the data array. It then attempts to find a natural grouping or clustering of the data sets based on a comparison of their image arrays. This comparison is done on a cell by cell basis and each data set image is compared to all others.

The corresponding cell entries of two image arrays are said to be equal if the difference between them is less than some delta, which can be determined by the user. If the entries are equal, the cells are said to match and a value of 1 is assigned to this cell comparison. If the entries are not equal then a value equal to the product of entries is assigned comparison. This is done for each cell within a particular column and the matching scores are added together. The score is then normalized by the maximum possible column score. This procedure is repeated for each column and scores are again added together and normalized this time by the number of columns.

The end result of this matching process is a number between 0 and 1 that gives the relative degree of similarity between two data sets based on their images. If each data set is compared to every other one, the results can be arranged in a square array whose size depends on the number of data sets. The diagonal elements of this array are all unity since these elements represent comparisons between the same data sets.

Image begins with the first element of the first column and places the corresponding data set into class 1. It then searches down the column and assigns any other data set to class 1 that has a matching score above some user selected threshold. It then moves on to the first element of the second column and assigns the corresponding data set to class 2. Again it searches down the column and assigns data sets to class 2 that meet the matching threshold. This results in the construction of a class for each data set contained in the original data array.

Now some of these classes may have only a single data set while others may have many. However, in general, the same sets will often be assigned to different classes. This has nothing to do with the data sets themselves, it is simply a function of the way the initial clustering is done. At this point Image goes back starting with class 2 and insures that the data sets in that class have not appeared in a previous one. If it has, the subroutine eliminates that data set from all subsequent classes. In this way the number of classes is significantly reduced and the data set membership in each class is unique.

The operation of Image in the classification mode is more straight forward. The subroutine again receives a data array from Poet, but in this case the array contains only one data set - the one the program is trying to classify. The subroutine simply makes an image of the data set as before and compares it to a data base of array images developed during an earlier clustering run. It then selects from the data base, the image giving the

best match to the current data set. Since each image in the data base was assigned to a class during the clustering operation, the routine simply assigns the current data set to the same class.

In its final form, Image will make a call to another subroutine called Motion. The purpose of this routine will be to specifically identify non-stationary processes. Examples of these are given in figures 7 and 24. In figure 7 we see slow but continuous rise in the overall pattern, while in figure 24 we see a discontinuous change in the pattern (perhaps indicating a misassociation). Essentially Motion will scan the image array to identify these data sets.

4.3.3 Hysteric

The subroutine Hysteric accepts the data array file from Poet and does clustering and classification based on the distribution techniques discussed in section 2.

In the clustering mode, Hysteric considers each file in the data array and normalizes the time axis of each file so that the data stream is of unit length, i.e., normalized time. These are outputted with an "N" prefix to each file's name for viewing and comparing. We do this because if we are not estimating the motion of the body, the time interval between samples has no significance.

The subroutine proceeds to calculate the histograms, normalizing them to unity to get the probability distribution function or PDF. It then calculates the cumulative probability distribution or CPD for each file. Each PDF is directed to a file with an "H" prefixed to the track file ID number, and each CPD is sent to a similar file with a "C" prefixed to the ID number.

Each histogram is filtered to remove the zero bins that may occur between bins with non-zero entries. This is done using either a median or moving average filter. The reason for doing this is that the RCS is really a continuous function of time. The fact that the histograms may have gaps or holes is really only due to the fact that the radar has discretely sampled the RCS function and thus can miss some samples within the possible range of RCS values.

It should be noted, that depending on the sampling interval, the same object could produce a histogram with holes at certain RCS bins which would be different from the holes due to a slightly different sampling time set. This is a critical problem with using PDF/histogram patterns and should be investigated further.

The subroutine next calculates the average value or "piston", "tilt", RMS residual, and peak for each RCS file. A vector of these values (called PRT) is then constructed as a possible pattern vector, with an option of including the peak value. This particular pattern

analysis can be invoked by setting certain parameters in an input file which is called by the subroutine Hysteric.

The subroutine now forms a set of pattern vectors for the RCS histories. Generally, a linear combination of PDF and PRT patterns is used to do the clustering. The user inputs a value, "wt", which can range from 0 to 1, with 1 favoring only PDF patterns and 0 favoring only PRT patterns. A value of wt equal to 0.5 favors each equally. When clustering is done, a value of the similarity measure is calculated for both PDF and PRT patterns for a given candidate relative to class. A weighted average of the two similarity measures is then used to arrive at a final similarity measure to test class membership.

As a last step, the routine then clusters the data sets and outputs the file called Cluster.dat.

4.3.4 Spectrum

At this point Spectrum also has only one mode of operation, although this subroutine will be expanded in the future. It accepts the data array from Poet and computes the Fourier transform for each data set. However, unlike the Fourier results presented in section 3, Spectrum does require that that the data interval be uniform.

Currently this routine only checks for RV-like objects, but we plan to expand this to include a check for all object types.

Section 5
Classifier Processing Results

5.1 Introductory Remarks

Once again, it is useful to review what we have done and what the objectives are. We stress that the basic idea is to identify patterns within the RCS data. Having a large set of track files of individual objects, we have been able to examine, first in an informal manner and then more quantitatively, the different types of patterns that we found to exist in our PAVE PAWS data base.

In order to emphasize various characteristics of the data patterns, we have developed a number of data processing techniques that resulted in simplified representations of the data. In section 3 we reviewed the results of using MATLAB to analyze some of the particular features of the individual RCS data sets. We found that many of the visual differences in the data sets could indeed be captured easily using these techniques. Moreover, these techniques and others could be coded as formal algorithms.

In section 4 we described the structure and operation of a prototype classifier (Poet). The program is built around four major subroutines. The subroutines are simply the algorithmic formulation of the data processing techniques discussed previously in section 2.

Finally in this section, we now present the results of applying our classifier to the data. First we consider the clustering mode. In this case we use the classifier to search through the data base and select those objects that meet the user selected requirements. The program then attempts to cluster the objects into groups based on patterns identified in their RCS time histories. This establishes the various *pattern classes*.

We then look at the results of running our program in the classifying mode. With the pattern classes in place, we consider new data sets and attempt to identify patterns within those sets and match them to the patterns classes already established.

Before we present our results, an important point needs to be stressed. Since we have not yet received truth data requested from the Navy, we have not verified the results of our classifier. Of course some objects are most certainly tanks and others should be RVs, but we expect that future research and analyses, using truth data, will substantiate these phase 1 research results.

Our work (based on our understanding of the scattering physics) has been encoded into the first subroutine (Prim) of the classifier. In this case the various thresholds on the values of the subset means and standard deviations determine the object classification that the subroutine assigns to the data set. This assignment or *object class* will be referred to

as we present the results of the clustering process. However it should not be construed as a "grading" of the clustering process, but rather as simply a way of organizing our results.

Our aims in this section are modest. We use Prim to establish the object class for all of the data sets. We then show the results from a few clustering attempts. The clustering is performed independently of the object classification and we grade the clustering results on the basis of the number and naturalness of the grouping. By this we mean that the data sets clustered together should look alike in some respect and that we should not end with as many groups as data sets.

5.2 Data Set Processing

In all of the clustering cases considered below, the number of data sets used was 46. We start below with output from Prim.

5.2.1 Prim Results

Currently the thresholds are hard coded into the Prim subroutine. Thus the object class was the same for each of the clustering runs. The output form Prim is show below in table 3.

6145	2	.818	-.068	.292	.204	.456
6149	3	.677	.455	-.136	.094	-.126
6153	1	.355	.197	2.683	-.008	.071
6154	3	.611	.319	.300	.293	.409
6168	3	1.000	-.043	-.678	.094	.129
6178	1	.676	.137	.163	.003	.094
6193	1	.643	-.682	.619	-.020	.134
6208	2	.429	.511	-.671	.648	.733
6221	1	.471	-.401	-.332	-.135	.166
6224	1	.565	-.415	-.430	-.089	.059
6234	1	.500	-.348	-.349	-.111	.154
6344	2	.850	.111	-.109	.449	.619
6347	1	.370	-.161	-.026	-.154	-.045
6351	2	.800	.192	-.355	.715	.797
6354	3	.280	-.172	-.523	-.181	.035
6406	3	.857	1.417	2.491	.566	.195
6414	1	.417	.667	2.177	-.053	-.189
1229	2	.632	-.183	.381	-.208	.161
1228	3	.821	.915	.549	.429	.152

Table 3. Object Classification from Prim

1231	2	.667	.016	-.357	.239	.473
1234	1	.449	.177	.073	.231	.127
1249	3	.857	-.380	-.402	-.341	-.545
1256	3	.800	-.378	-.451	.058	.044
6426	1	.250	-.099	-.541	.336	.433
1262	4	1.000	1.115	7.248	-.034	-.144
1275	1	.625	-.059	-.177	.049	.020
6429	1	1.000	.718	-.494	.519	.544
1293	3	.533	.829	1.535	.241	.180
2348	2	.556	-.139	-.087	.004	.378
2349	3	.895	1.066	.775	.666	.367
2364	3	.500	-.169	-.733	.259	.333
2393	3	.857	-.150	-.607	-.149	-.420
2394	3	1.000	-.356	-.960	.067	.117
1370	4	1.000	-.611	-.285	-.724	-.767
2424	1	.867	.021	1.117	.015	.126
2448	3	.833	1.466	2.661	.346	.122
2499	1	.857	-.755	1.258	-.369	-.108
2521	3	.300	-.349	1.184	-.118	-.277
2530	1	.333	-.712	-.472	-.353	-.338
1490	3	.615	1.486	3.003	.553	.584
2582	1	.750	-.399	-.478	-.277	.148
2594	1	1.000	-.273	.977	.301	.784
2597	1	.765	-.486	-.603	-.472	-.349
2609	1	.857	-.434	-.510	-.131	.123
2631	1	1.000	-.376	-.523	.385	.719
2637	1	.750	1.862	6.489	.680	.774

Table 3 (Continued)

The first column of table 3 gives the object identification number and the second column gives the object classification as determined by Prim. The fraction of subset values satisfying the classification assignment is shown in column three. This fraction provides a measure of confidence in the Prim classification (see section 4.3.1 for a review of the Prim subroutine).

This confidence factor can be small, since there are four possible classes and the assignment is made according to whichever class has the most points. One should also note that in the case of object 6426, the fraction is only .250. In this case the subset points were evenly distributed among the four classes, and, the subroutine simply defaults to class 1.

The remaining columns give the output from the H_mom and Correl subroutines contained within Prim. This includes data concerning values of the higher moments of the

data sets and the degree of correlation between first and second moments. This information has not been particularly useful up to this point and will not be discussed further.

As a reminder, the four types of object classes are shown below in table 4.

Class 1	RV
Class 2	Tank
Class 3	PBV
Class 4	Fragment

Table 4. Object Class Types

5.2.2 Image Results

In the clustering mode, the Image subroutine clustered the data sets into 12 pattern classes as shown below in tables 5 and 6.

Pattern Class 1		Pattern Class 2		Pattern Class 3	
Object ID	Object Class	Object ID	Object class	Object ID	Object Class
6145	Tank	6149	PBV	6153	PBV
6344	Tank	6178	RV	6193	RV
6354	PBV	6406	PBV	6224	RV
1229	Tank	6414	RV	6221	RV
1231	Tank	1228	PBV	6234	RV
2348	Tank	1234	RV	6426	RV
		1293	PBV	1275	RV
		1490	PBV	6429	RV
				2424	RV
				2499	RV
				2521	RV
				2530	RV
				2582	RV
				2597	RV
				2609	RV
				2631	RV
				2637	RV
				6347	RV
				2364	RV

Table 5. Clustering of Data Sets by Image into Pattern Classes

The remainder of the data was clustered into pattern classes with only one or two data sets each. The explicit grouping is shown in table 6.

Pattern Class 4	Pattern Class 5	Pattern Class 6	Pattern Class 7	Pattern Class 8	Pattern Class 9	Pattern Class 10	Pattern Class 11	Pattern Class 12
6154	6168	2393	6208	2394	6351	1256	1249	1262
		2448				2349		1370

Table 6. Clustering of Data Sets by Image into Pattern Classes

In table 6 all of the objects were classified by Prim as PBVs with the following exceptions. Objects 6208 and 6351 were classified as tanks and objects 1262 and 1370 were classified as fragments.

A few remarks are in order. First we note that the pattern class number is arbitrary and is simply used as a method of labeling. In particular, they bear no relation to the object class numbers.

In terms of what we see in the tables, the overall clustering looks reasonable. In general, the tanks would appear to all look very much alike, i.e. there is essentially only one tank pattern with the exception of two singular cases (objects 6208 and 6351).

Moreover, one could say that the RVs appear to map into only two pattern classes, at least according to this subroutine. However, we need to be careful in our assertions. We have not verified the object classification with truth data, so all we can really say is that a significant portion of the data not in pattern class 1, can be put into pattern classes 2 and 3. On the other hand, we can feel fairly confident that of objects in pattern classes 2 and 3 many are likely to be RVs.

A somewhat disappointing aspect of the clustering is the rather large number of small or singular groupings. If the clustering was in some sense optimal, that is if one was assured that the grouping represented the minimum value of some test function for example, then the singular groups would not be a source of concern. Indeed, we would simply state that those singular groupings were formed only because the data set patterns really were different from any other. That is we had a number of singular and unique patterns. However, in this case, we have no particular reason to assume that the clustering is optimal. This issue will be addressed again in section 6.

Figures 30 and 31 (pages 58 and 59) show the RCS time histories for the objects in pattern class 1.

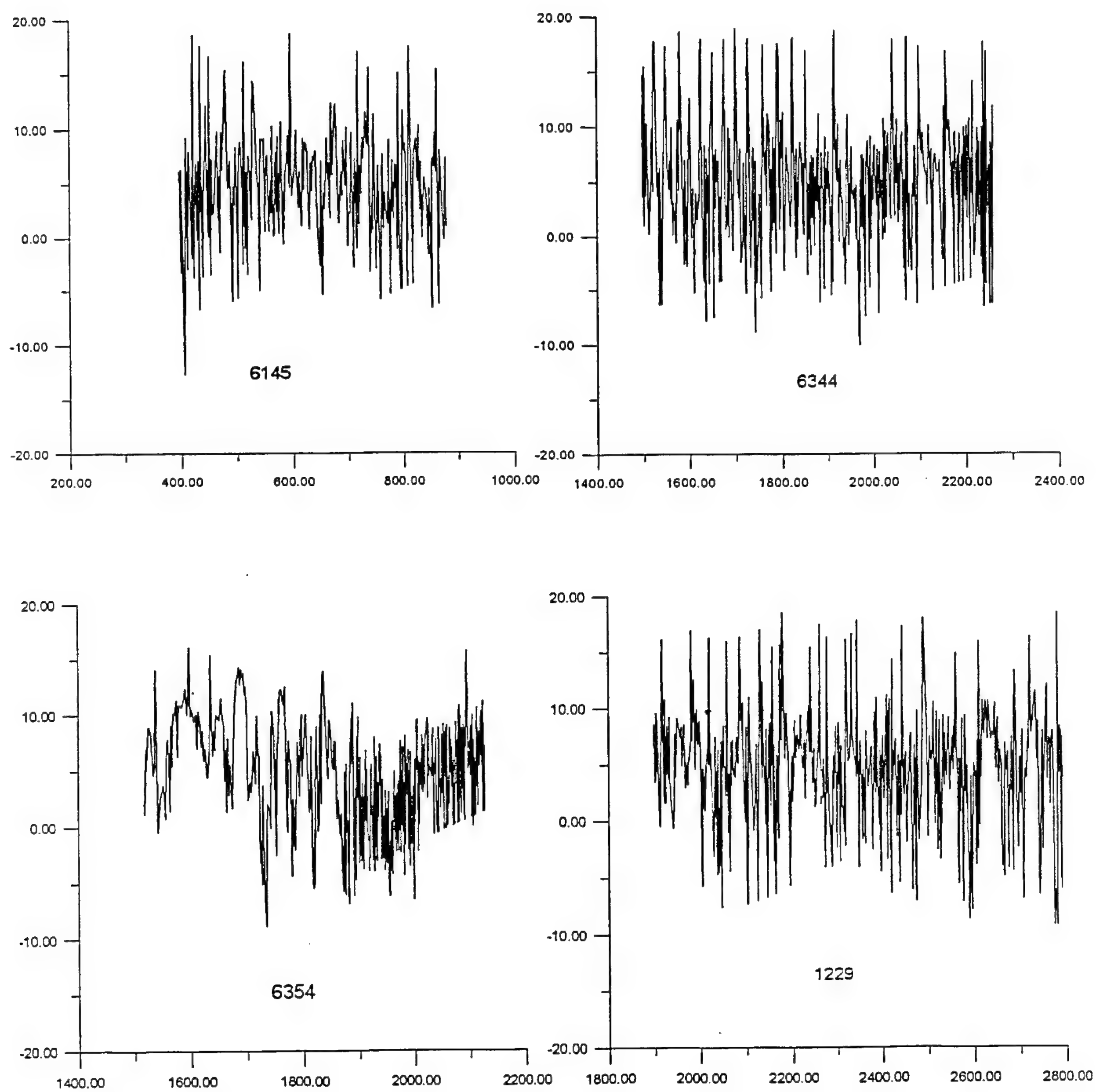

Figure 30. Objects in Pattern Class 1 (Image)

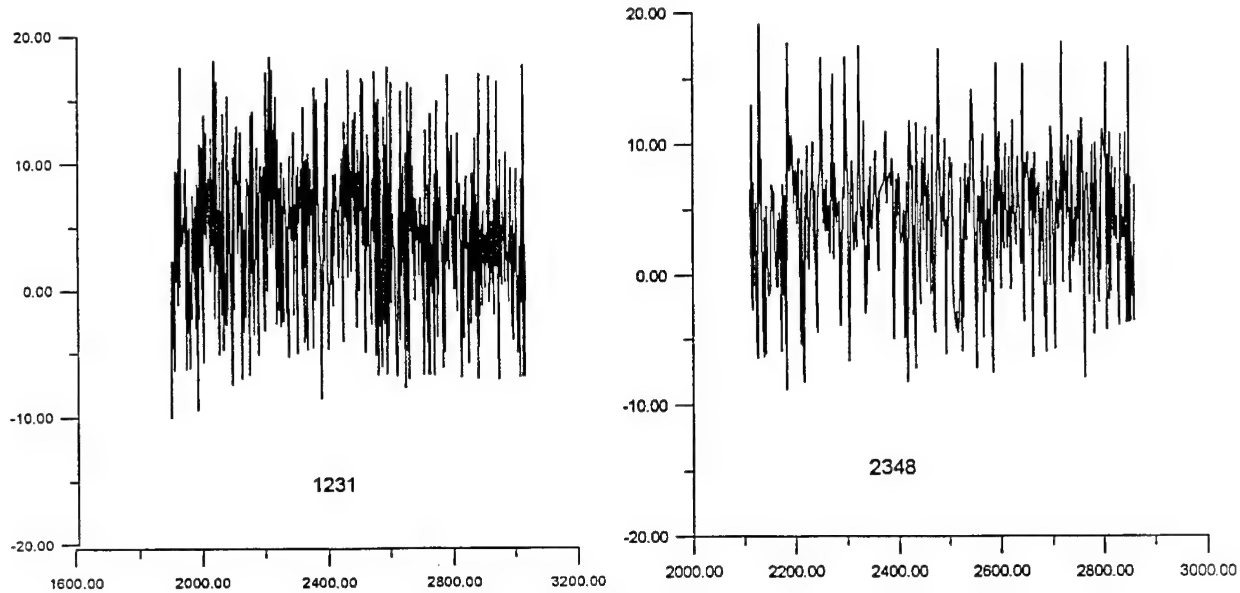

Figure 31. Objects in Pattern Class 1 (Image)

In general the class 1 clustering looks reasonable, in that all of the plots show the same kind of "spikey" variability centered around a relatively large mean. However, object 6354 represents a slight shifting from the established pattern. This is a rather common problem that occurs in clustering analysis. There is a tendency for the pattern class to expand as more data sets are examined. The reason for this is that occasionally data sets which match the pattern class in only a marginal way are accepted into the class. Once in the class, they tend to attract other data sets whose match to the original pattern is even worse. Eventually, the pattern class can become rather meaningless in the sense that almost any data set can get in. Of course the way to prevent this is to demand a closer match before a data set is placed into a group. However, this must be balanced with measures to prevent creating a large set of pattern classes with only insignificant differences between them.

One should also note that the Image subroutine is not particularly sensitive to the data rate. Thus if the wide "lobe" structure seen in the first part of the RCS plot of 6354 is simply a function of a low track rate, then the placement of this object in pattern class 1 is seen to be a relatively good choice.

Figures 32 and 33 (pages 61 and 62) show the time histories for data sets in pattern class 2. This class contains both PBV and RV-like objects. These data sets also show a "spikey" pattern, but in addition they also display a general increase (or decrease) in the DC or mean level of the RCS amplitude as a function of time.

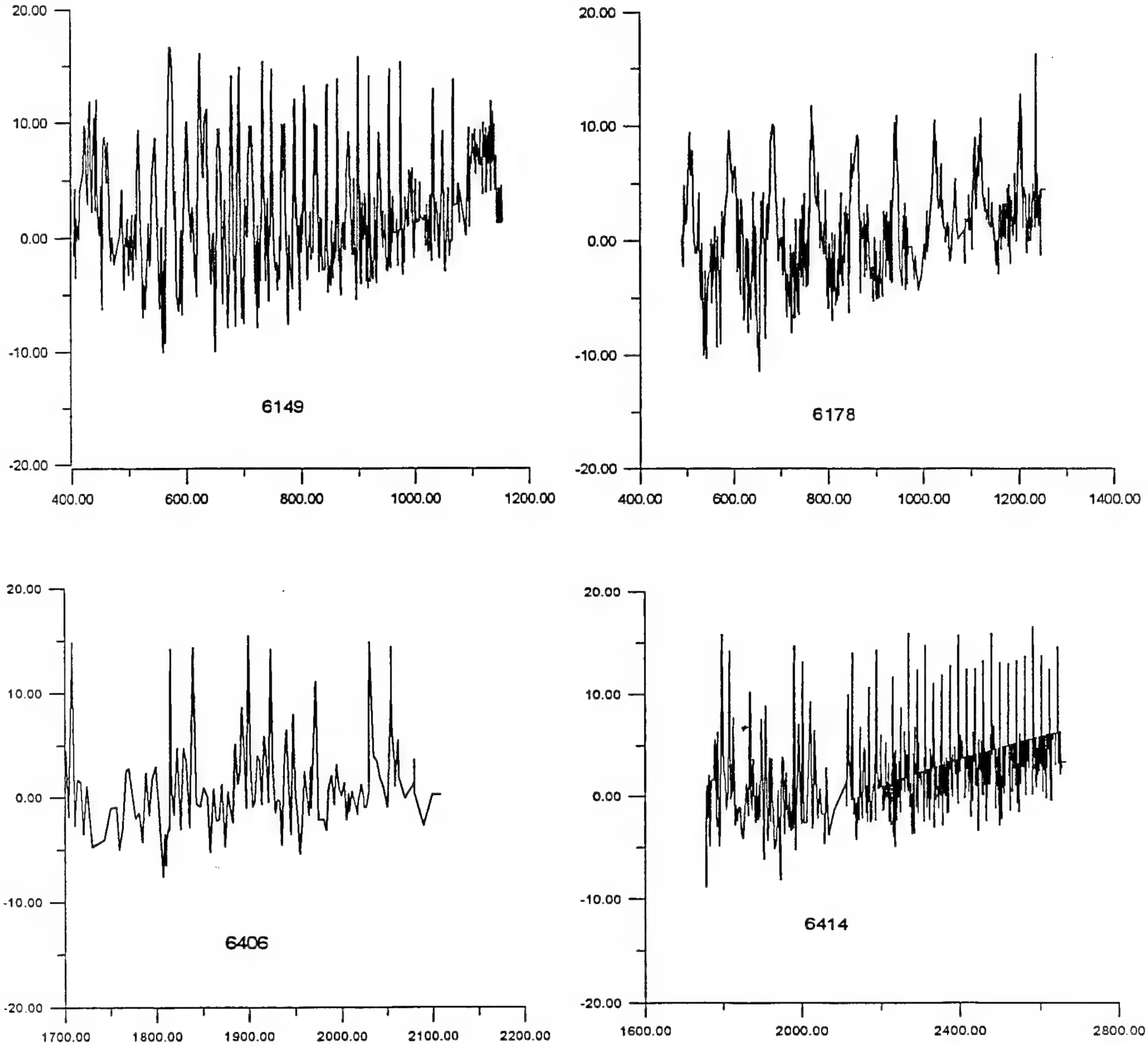

Figure 32. Objects in Pattern Class 2 (Image)

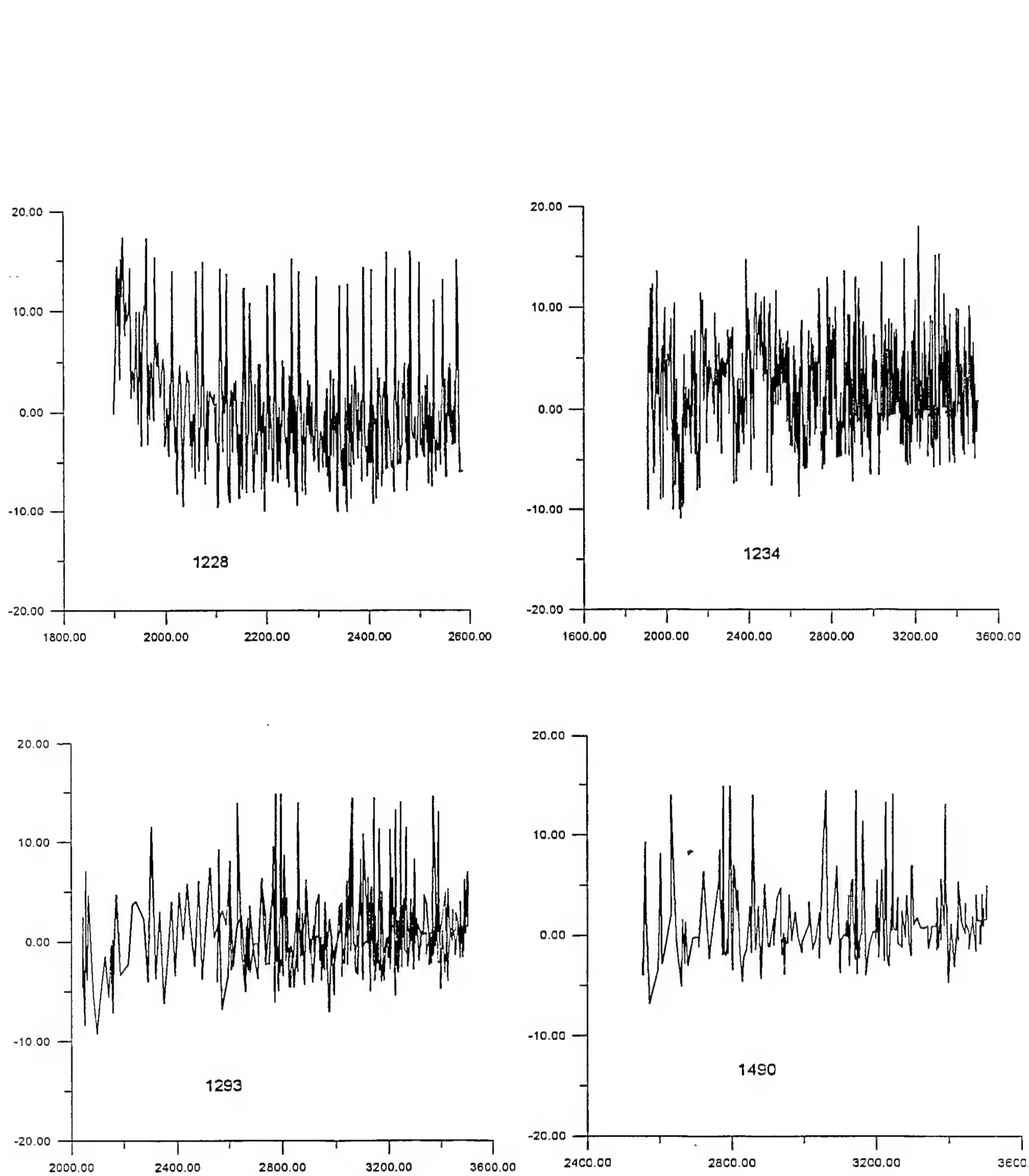

Figure 33. Objects in Pattern Class 2 (Image)

The next set of figures, 34 through 38, (pages 64-68) give the RCS time histories of the data sets which were clustered into pattern class 3. All of the data sets in this pattern class were classified by Prim as RV-like. This class turned out to be the largest class with 20 members. Again, in general, the grouping looks very good. Most of the data sets show a pattern characterized by a small mean and variance, plus a systematic rise in the overall pattern as a function of time. Note also that Image is not particularly sensitive to small numbers of spike-like amplitudes, as seen in objects 2637 and 2424 (pages 65 and 66).

This is particularly obvious in the case of object 6153 (see figure 37, page 67). In this case, it may appear as if the routine made a mistake. But, in fact, the matching of this data set to pattern class 3 is quite good. If we look closely, we actually find that a large majority of points in this data set behave much the same way as do points in other sets that make up this class. However we are naturally drawn to view the spikes in the pattern.

The question becomes, how much significance should be attached to these types of features. This question might be best answered once truth data is available. However, given the fact that this whole approach might ultimately represent a first cut in the target discrimination process, we probably do not want to leave any RV-like object out of consideration; as might be the result if too much emphasis is placed on a small number of prominent features.

A final remark relative to this pattern class is that again we see some spreading. Objects 1275, 2364 and 6347 (figure 38, page 68) appear significantly different from the other data sets and perhaps should be put in a class by themselves.

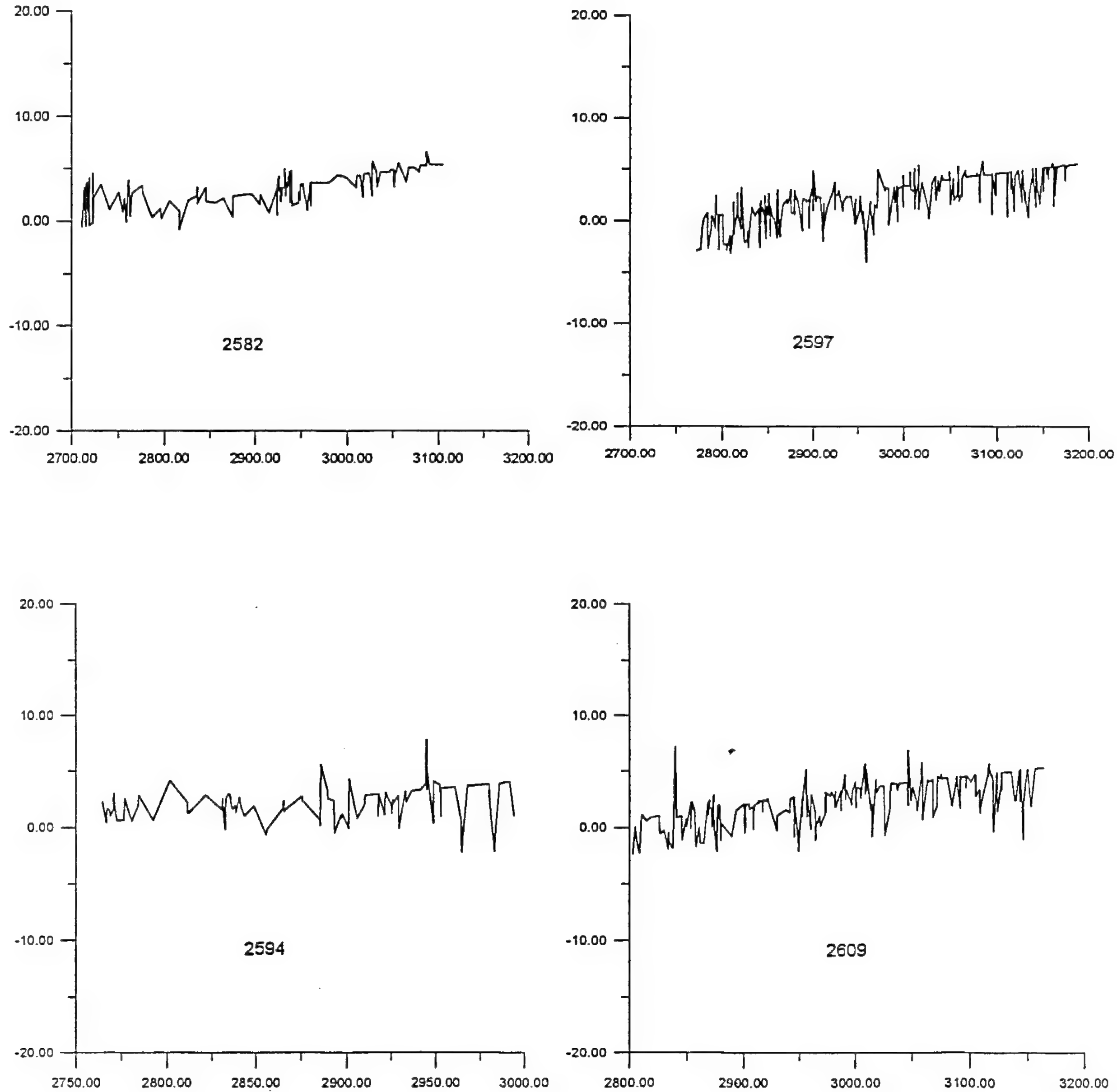

Figure 34. Objects in Pattern Class 3 (Image)

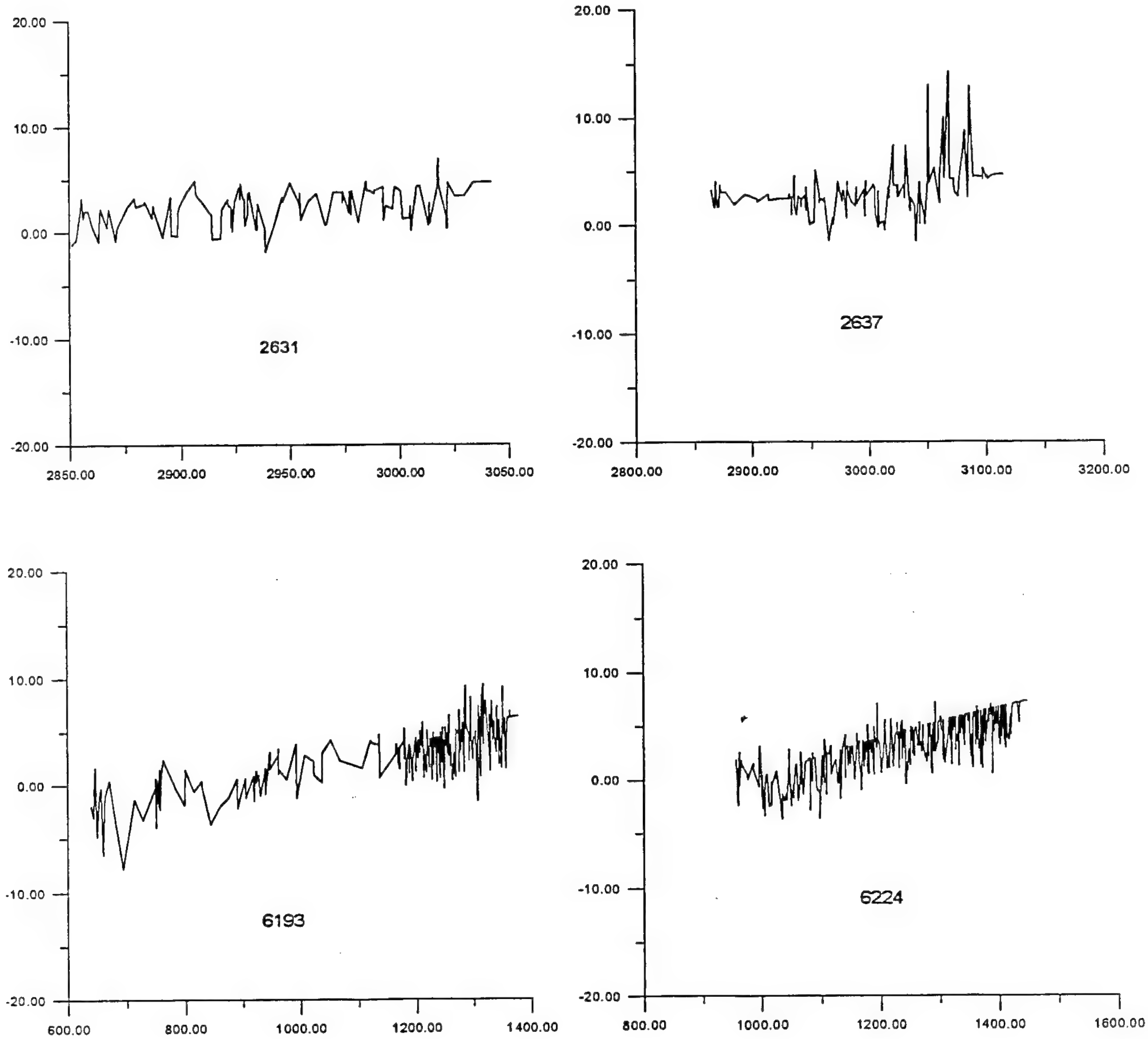

Figure 35. Objects in Pattern Class 3 (Image)

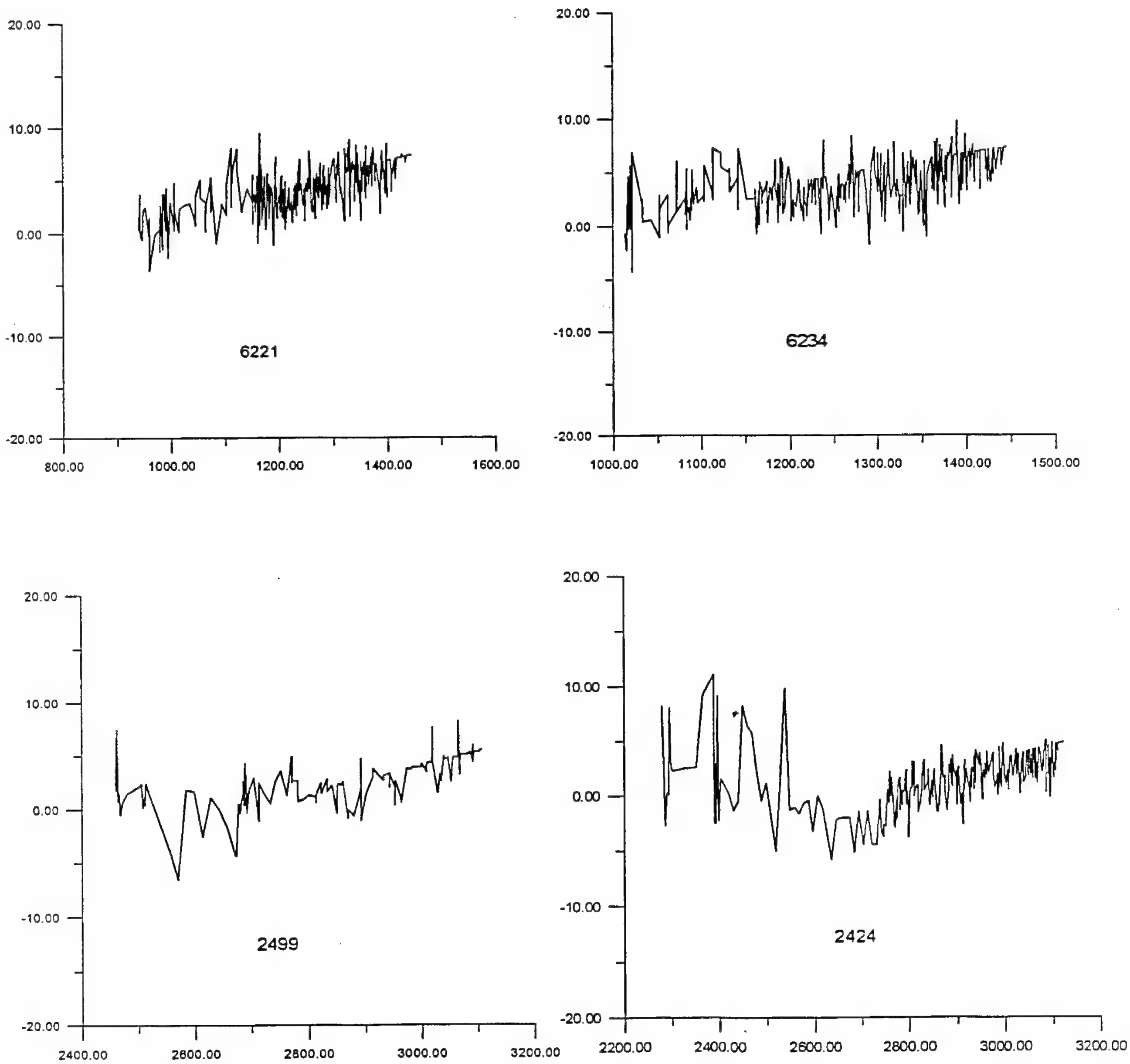

Figure 36. Objects in Pattern Class 3 (Image)

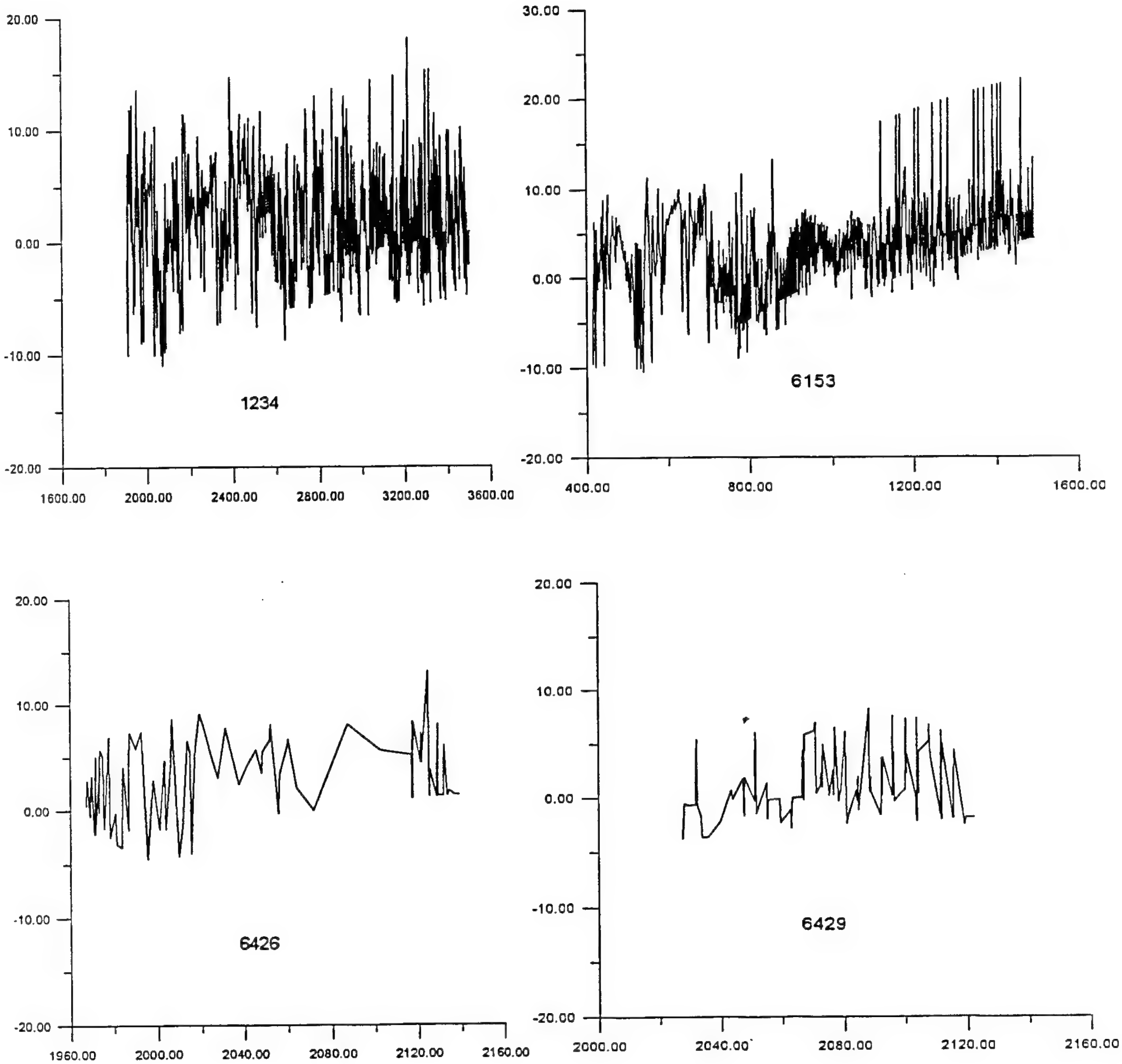

Figure 37. Objects in Pattern Class 3 (Image)

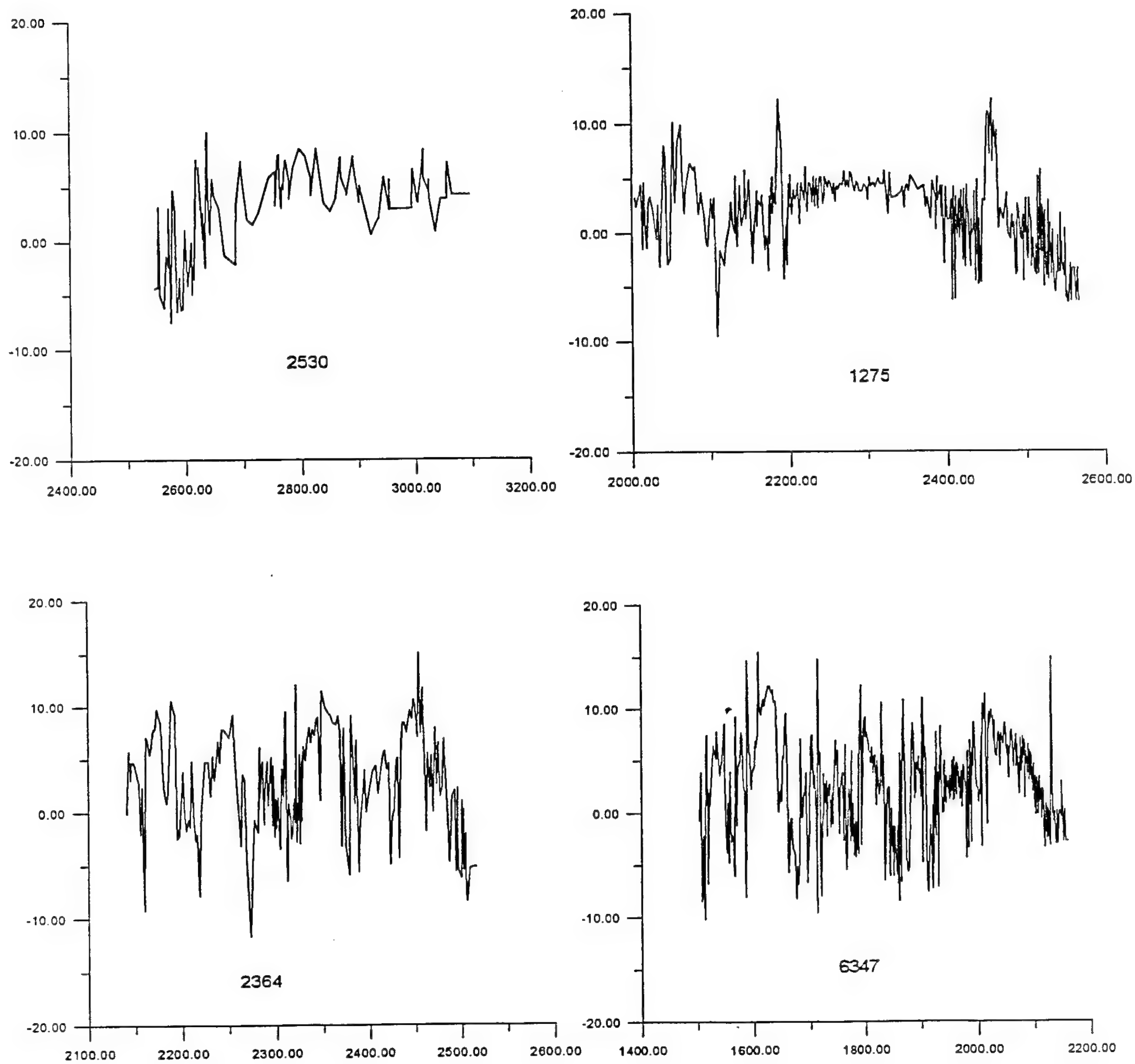

Figure 38. Objects in Pattern Class 3 (Image)

The final set of examples are shown in figure 39 (next page). These plots represent a sampling of data sets from those pattern classes containing only one or two members.

For example, objects 1262 and 1370 were placed by themselves into pattern class 12, as their patterns are quite distinctive. These were classified as fragments by Prim based on their small RCS.

Objects 6208 and 2394 were put into singular classes 7 and 8, respectively (see page 57). This appears reasonable, as they both present patterns that appear quite different from what we have in the previous examples. Prim classified 2394 as a PBV and 6208 as a tank.

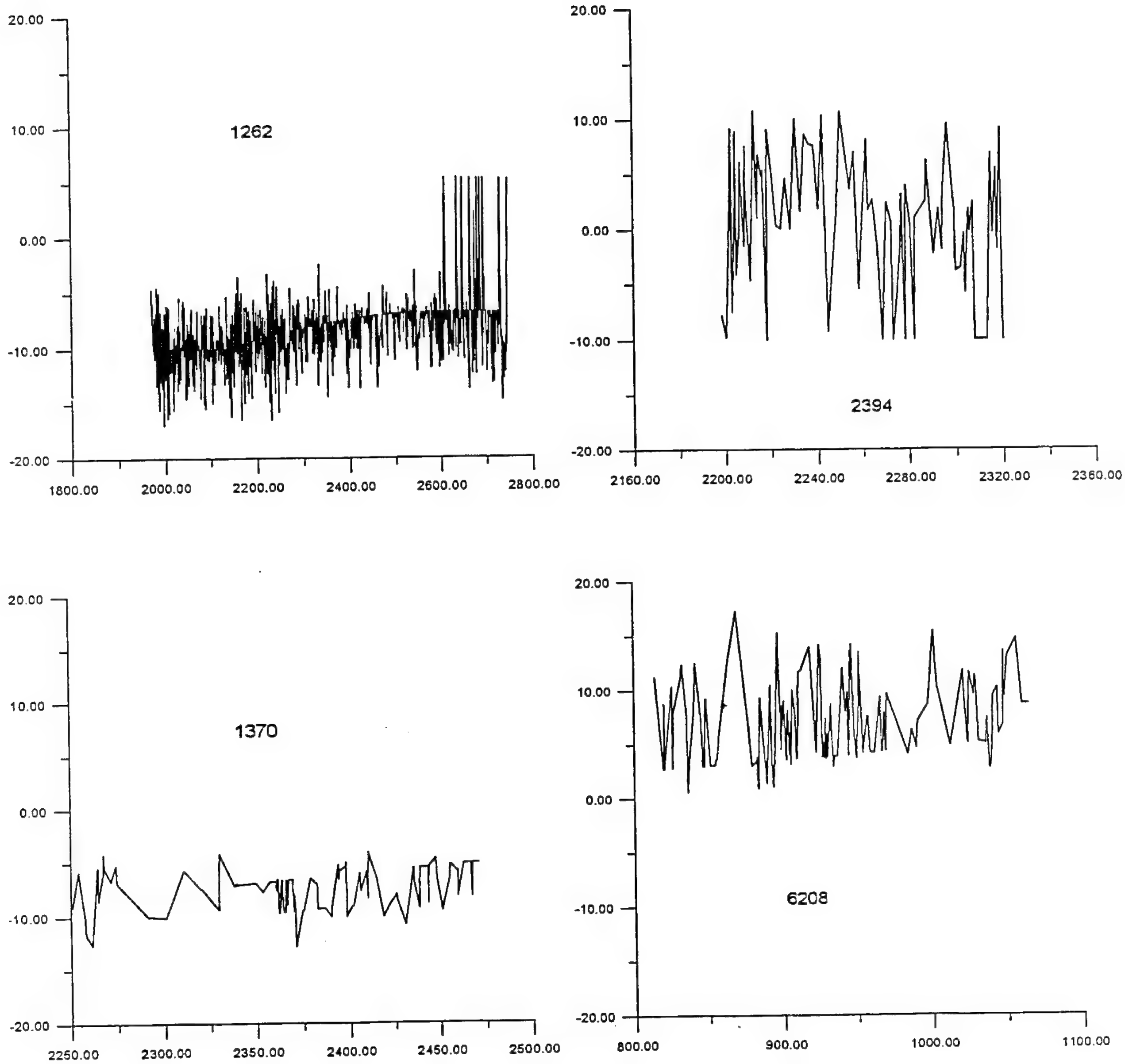

Figure 39. Objects in Small Pattern Classes (Image)

5.2.3 Hysteric Results

The results from Hysteric are similar to those derived from Image, in the sense that many of the objects that are clustered together by Image are also clustered together by Hysteric. But there are some interesting differences. Many of the objects that were classified by Prim as PBVs and spread over many pattern classes by Image, are now consolidated into principally one pattern class with a few others being clustered together with the tanks to form another class. The RCS time histories for a representative sample of these objects are shown in figures 40 and 41 (pages 72 and 73).

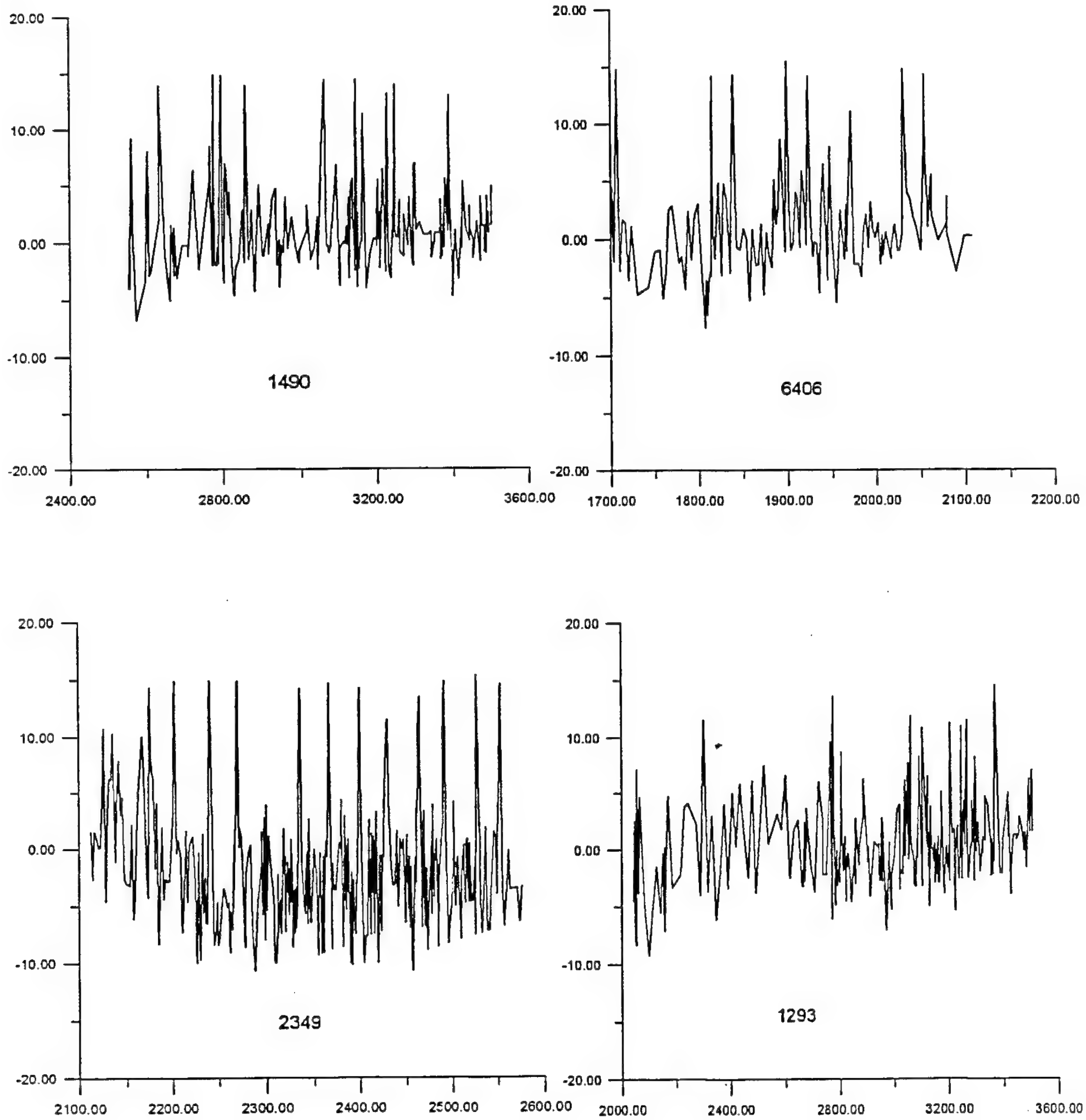

Figure 40. Objects in PBV Pattern Class (Hysteric)

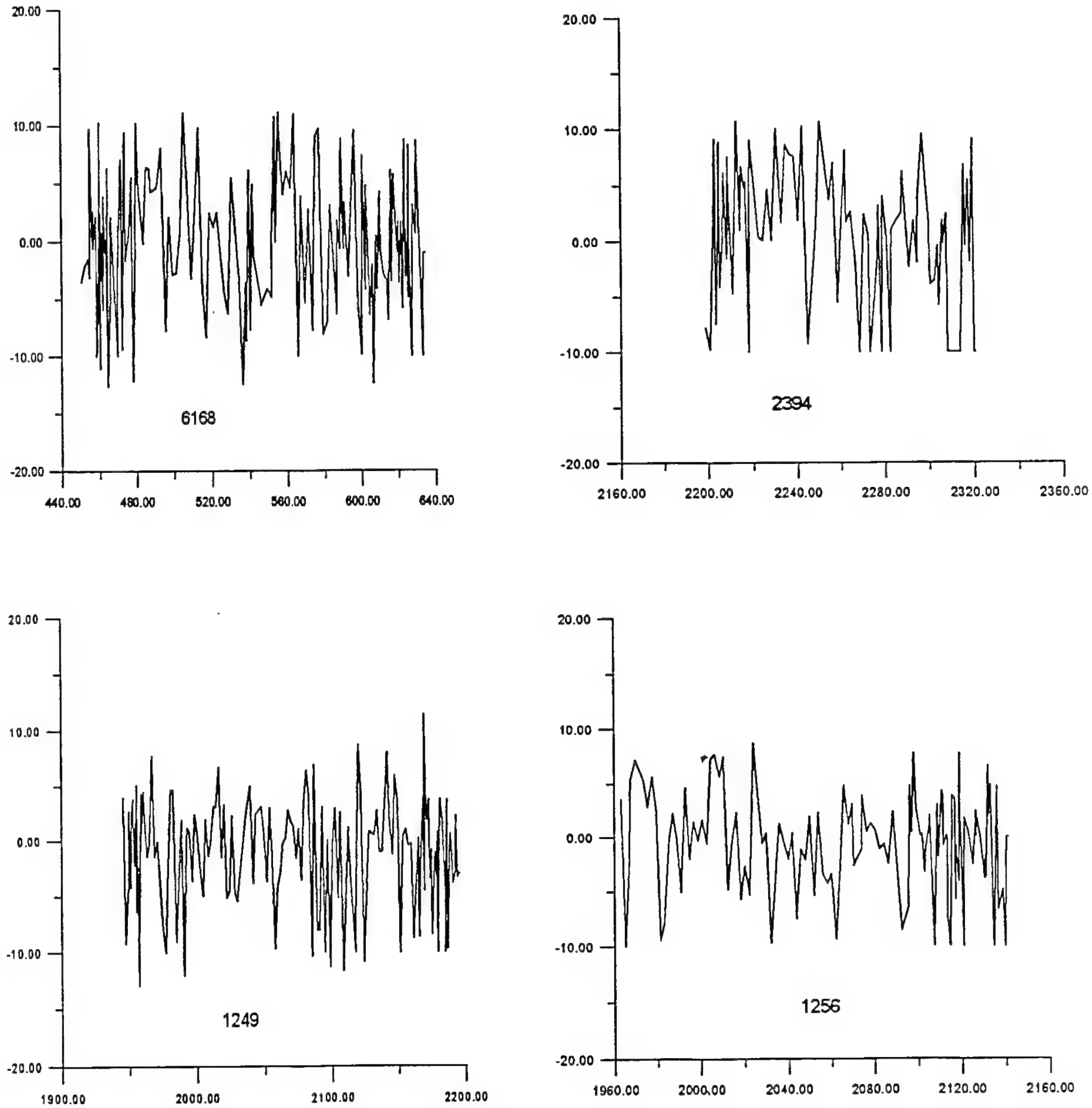

Figure 41. Objects in PBV Pattern Class (Hysteric)

The second interesting difference concerns the RV-like or object class 1 objects. Hysteric distributed those objects over 3 major pattern classes, which as we saw from the Image results might not be a bad thing to do. However, Hysteric also placed some RV-like objects into its tank pattern class.

At first, this would appear to be an incorrect assignment. But again since the object classification has not yet been verified with truth data, one can not say if this placement is incorrect or not. In any case, the RV-like objects (6347, 1234 and 2364) that were placed in the tank pattern class are not particularly RV-like, at least from a visual point of view. The RCS time histories for these objects are repeated in figure 42 on the next page and should be compared to the RV-like patterns displayed, for example, in figure 34 (page 64).

In a real sense, this is just the kind of result one would like to see. After all, Hysteric and Image do pattern matching and clustering in somewhat different ways and really consider different aspects of the pattern. Where the differences should be important, are in those cases where the data set represents a borderline case. In this situation, the disagreement between the two subroutines can be flagged and the program can either attempt to resolve the disagreement (e.g., utilizing the Poller subroutine, which we plan to develop later in our research) or at least identify the object as a problem or borderline case, perhaps requiring special attention.

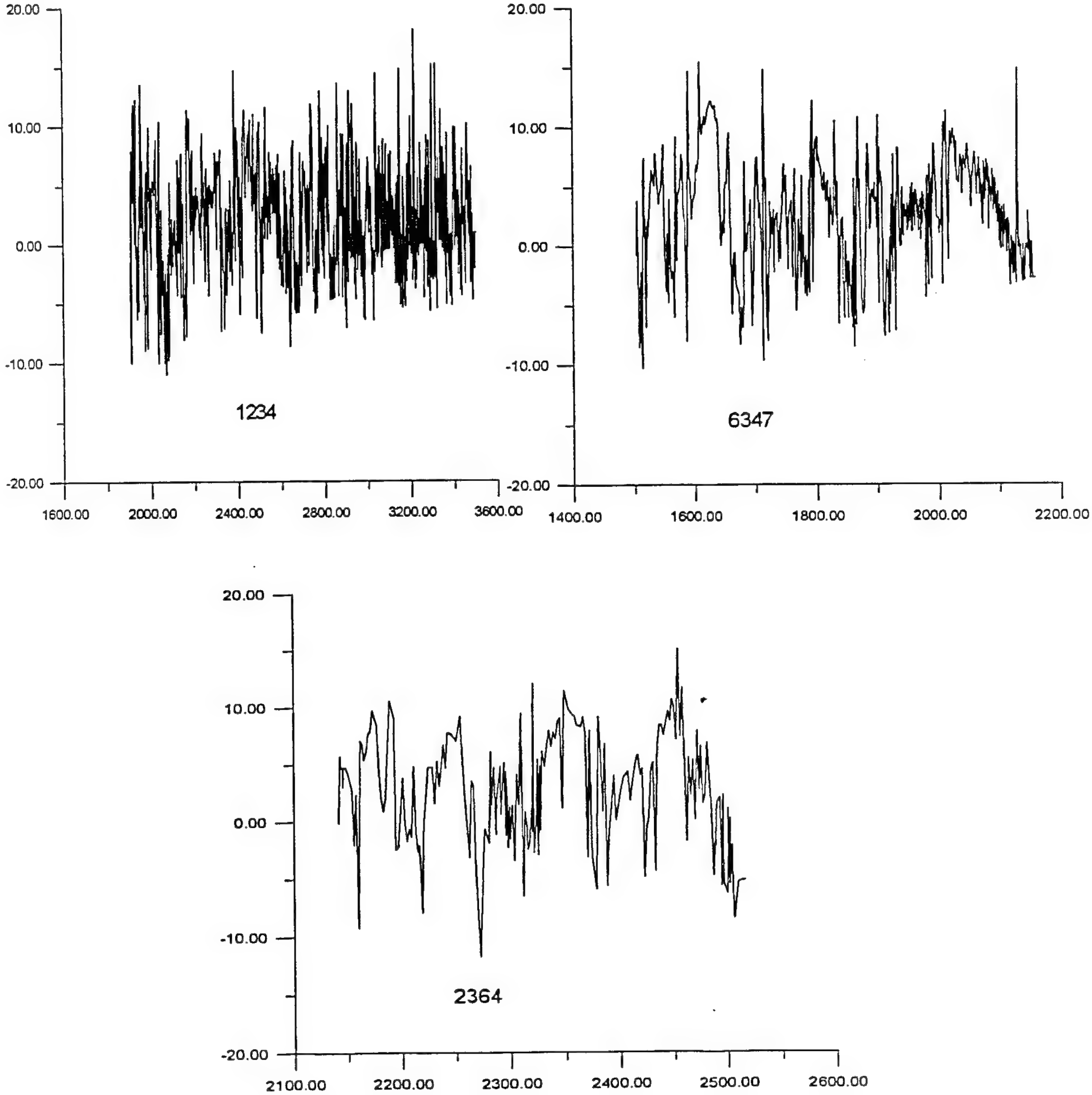

Figure 42. RVs Placed in Tank Pattern Class (Hysteric)

5.2.4 Spectrum Results

Currently this subroutine only performs a check for RV-like objects. It does this by calculating the percentage of coefficient points contained within in a circle of a given radius. This radius threshold was based on only one RV-like data set and so the results from this subroutine are only tentative. However, it did identify the following data sets listed below in table 7 as RV-like.

Object ID	Spectrum Classification	Prim Classification
6168	RV	PBV
6224	RV	RV
1256	RV	PBV
1262	RV	Fragment
6429	RV	RV
2393	RV	PBV
2394	RV	PBV
2582	RV	RV
2594	RV	RV
2597	RV	RV
2609	RV	RV
2631	RV	RV
2637	RV	RV

Table 7. Comparison of Spectrum and Prim Object Class 1 Assignments

The results are similar to those derived from Prim with some obvious differences. Spectrum designated a smaller set of objects as RVs and included some objects which were classified differently by Prim (see table 3, page 54).

Object 1262 (fragment) may have been included as an RV, since Spectrum currently does not have a lower threshold on the radius length. The reason why some of the PBVs were classified as RVs is unclear. However, it may simply be because we need to refine our radius thresholds based on analysis of more data sets.

5.3 Initial Classification Test

We have not had time to properly test this mode of the classifier. However we have made a few trial runs in the classifying mode, using the Image subroutine to determine the pattern class. These very preliminary results are encouraging, so we present a brief summary of them.

In order to evaluate the classifying performance, we chose not to use our entire RCS data base (i.e., group III) when we did the clustering to establish the pattern classes. Thus there remains a sizable number of track files, whose RCS time histories have not been examined or analyzed. The idea is to select objects from this portion of the data base and let Poet attempt to determine the object and pattern class for each one.

Determining the object class of course is trivial, since the thresholds are hard coded into the Prim subroutine. Determining the pattern class is a more interesting problem. Being able to take an unknown data set and place it into an established pattern class demonstrates that we have catalogued all of the pattern classes and that these patterns are not the result of some special circumstances occurring during a particular event.

This is an important point and should be emphasized. The pattern classes were established by using a LRID-like file constructed from data collected by the radar during two separate events. The classification test uses a second LRID-like file constructed from data collected from a third event.

To begin to evaluate the classification performance, we randomly selected three objects from the second LRID-like file. The only requirement was that the number of data points be at least equal to the minimum number selected for data sets used in the clustering mode. Again, it should be noted that we had not examined these objects before.

The objects and their classification are given below in table 8.

Object ID	**Object Class**	**Pattern Class (Image)**
8926	Class 3 (PV)	Class 2
9054	Class 1 (RV)	Class 3
9056	Class 1 (RV)	Class 3

Table 8. Results of Classifying Runs

Figure 43 (next page) gives the RCS time histories for these objects. The pattern assignments appear very good. For the case of 8926, Prim classified it as a PBV-like object and Image assigned it to pattern class 2. One should go back to figures 32 and 33 (pages 61 and 62) to convince themselves that the assignment to this pattern class is quite reasonable.

Objects 9054 and 9056 were both classified as RV-like objects and an examination of their RCS time histories indicates those as reasonable classifications. Moreover, the pattern class assignment (pattern class 3) appears correct as one can verify by consulting figures 34 through 38 (pages 64 through 68).

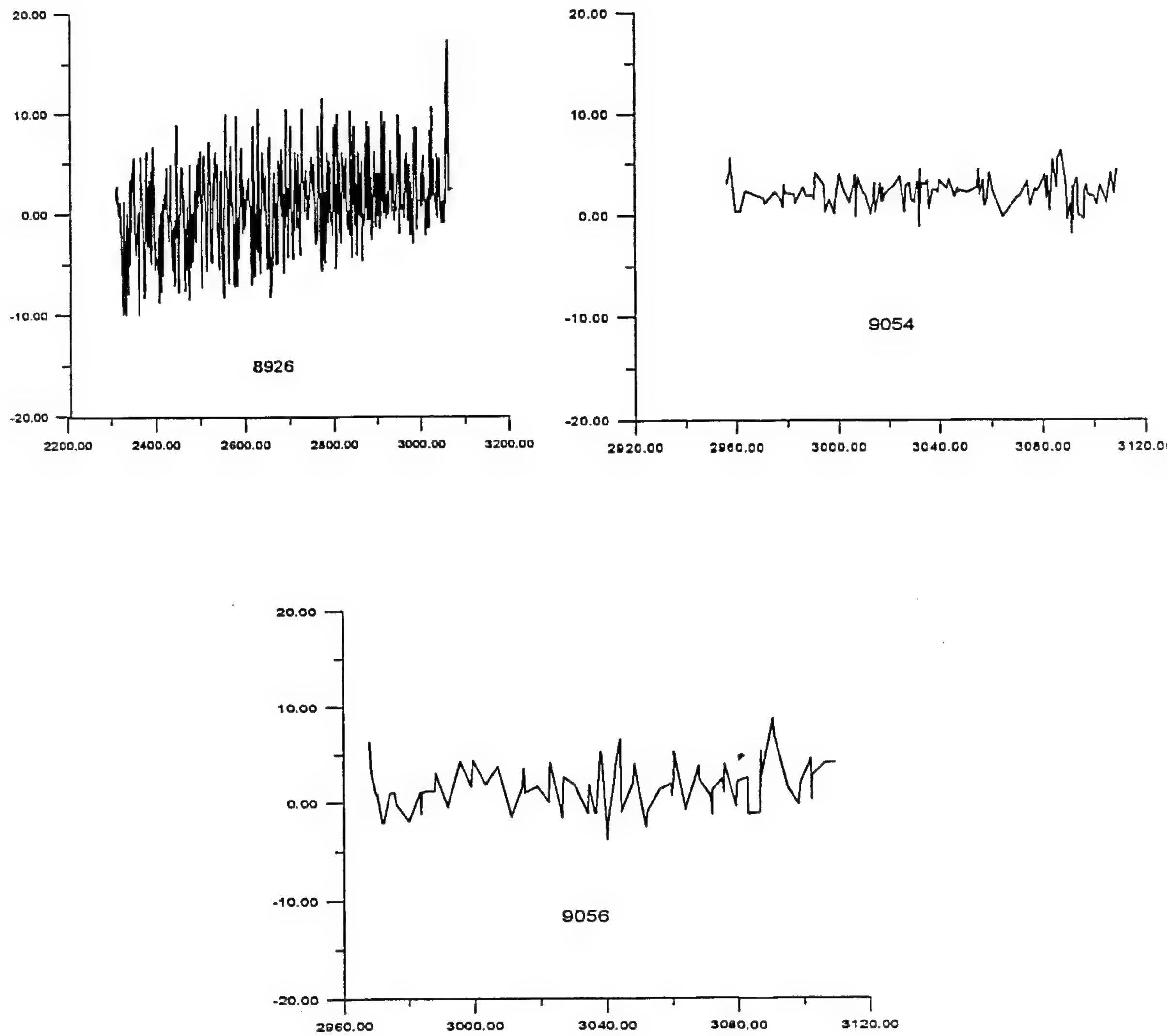

Figure 43. RCS Time Histories

Section 6
Summary Discussion and Conclusions

6.1 Summary of Data Survey Results

We can summarize our data survey results in the following way. First we divided our RCS data base into groups, based on collection dates. The data from groups I and II were processed through a primitive classifier which used the means and standard deviations of the component 20 point subsets of each data set. Based on some rough rules of thumb, these data sets were then assigned to one of four object classes, i.e., tank, RV, PBV and fragment.

We then incorporated most of the previously developed data processing techniques into a MATLAB framework. This allowed for an efficient and systematic search through the data base. It also represented the first step in establishing a way of emulating the pattern recognition based object classifier, which could be useful during further development and evaluation.

We then processed data from groups I and II through the MATLAB environment. This allowed us to test the utility of some of our data processing techniques. It has also demonstrated that by using these relatively simple techniques, we could identify other features in the data for which our primitive classifier was not particularly sensitive such as the frequency content and the distribution relative to the mean value.

This indicates that once we are able to assign patterns to objects, we have a methodology that can distinguish RVs from similarly sized objects. This hope is reinforced when we note that the patterns are repeated in events that occurred years apart.

6.2 Discussion of Clustering Results

In general the clustering results are quite promising. The patterns in the RCS data are prevalent and sufficiently distinctive such that most of the data sets can be clustered in a natural way into a reasonable number of pattern classes. This was clearly demonstrated in section 5. While we did note a number of singular pattern classes, over 70% of the data could be put into three or four pattern classes.

It should be emphasized, that the clustering process is a key step in this work because it allows us to construct the pattern classes. These pattern classes represent the key feature in the operation of our classifier. Thus finding the optimal or near optimal clustering is quite important.

Because our limited research has not yet necessarily resulted in the optimal clustering, having two pattern matching methods is particularly advantageous. For example, it takes some tuning of the Image subroutine parameters to get a good grouping for certain patterns. This tends to cause problems for other patterns. Thus we find that while the clustering results from the Image subroutine are generally quite good, it does tend to spread potential PBVs over many pattern classes. On the other hand, the Hysteric subroutine tends to cluster this object class into just a couple of groups.

The fact that the two clustering routines do not always agree is seen as a positive feature. This presents a way of identifying and analyzing border line or unusual cases.

6.3 Discussion of Classifying Results

One can not draw a conclusion based on just three test cases. However, these first results are very encouraging. In all three cases our classifier, Poet, was enable to assign or match the "unknown" data set to an established pattern class. Moreover, from a visual point of view, the matching appeared very good. Our future work will include substantive test cases to verify the performance of our classifier.

6.4 Future Efforts

There is of course an important remaining issue, which is the need to verify our results with truth data. Though we feel confident that we can eventually obtain at least a subset of this data, it is disappointing that we were unable to obtain it before the end of Phase I. While the results of this research project strongly suggest that this approach of doing object classification can offer a significant improvement over the current capability of the EWRs, a quantitative assessment of the pattern recognition approach can not be made until the truth data is available.

Thus one of the next steps in our work will be to use the truth data to label the data sets that are going into each pattern class. In this way we can determine if the idea of pattern classes is useful, in the sense of verifying that different kinds of objects really do map into different pattern classes.

Finding the best way(s) to do the clustering is another important near term effort. It is suspected that the approaches will be different for Image and Hysteric, since their methods for doing matching and clustering are different. In the case of the Image subroutine, one possible approach would be to establish a measure of the distances between pattern classes as a function of the individual members in the classes That is, the distance between classes would vary depending on which members were in which group. The best clustering might then be achieved when the distance measure between pattern classes was maximized, or equivalently, when the inverse distance was minimized. When the problem is posed in this fashion, a powerful approach for obtaining the solution is the method of simulated

annealing. For further details on this method, one should consult reference 1 identified on page 17.

In the case of the Hysteric subroutine, it is probable that the optimal approach to clustering will be determined from a closer look at the vector space formulation on which this method is based.

Finally, while the classifier program (Poet) is a real piece of functioning software, it is certainly not complete. In particular, the Poller subroutine needs to be written. The purpose of this subroutine is to settle any disagreement between Image and Hysteric. A disagreement would occur if the two subroutines placed the same object into pattern classes which were nominally associated with different object types. That is, one placed the object in a RV pattern class and the other placed the same object in a tank pattern class, as we saw occur in section 5 (page 75).

Subroutines such as Motion Detector also need to be written, while others such as H_mom and Correl need to be re-evaluated to determine if any useful information is being obtained from them. On the other hand, the Spectrum subroutine can probably provide more information, and thus its role within the classifier should be expanded.

Test and evaluation of the classifier is the another important step. In this case the effort will focus on ways in which the program can be tested in the "real" world. This could be accomplished either through testing at a PAVE PAWS radar site, or through a simulation conducted at an Air Force facility such as Detachment 25 in Colorado Springs.

6.5 Conclusions

The work accomplished during phase I of this research project has produced a number of solid results.

First, we found that different types of patterns exist in the EWR RCS data base and simple processing techniques can be developed to identify the various aspects of these patterns. As discussed in the report, the data was collected from events that occurred over a period of about two years and thus the patterns are not the result of special circumstances. Thus there is real value in attempting to discriminate on the basis of these patterns.

Second, it is also shown that most of the RCS data sets fall naturally into a reasonable number of distinct pattern classes. That is, the number of pattern classes is much less than the number of data sets examined and that the members within a pattern class do look alike.

Third, we have demonstrated that different object classes tend to lie in distinct pattern classes. By this we mean that objects that we believe are RVs and tanks have different

patterns. Thus being able to associate a pattern or pattern class to an object, i.e., doing discrimination on the basis pattern recognition, appears very possible.

Finally, a computer program was developed which, by using the various methodologies developed, can generate pattern classes and perform a discrimination function based on the assignment of RCS time histories to these pattern classes.

These results strongly suggest that further work in this area will be very valuable. We intend to continue this work and move toward an evaluation and test phase.